inside electronic game design

- Read about the history of electronic game design—and where it's going in the future!
- Exclusive interviews with 24 leading game designers!
- Learn how to launch a profitable career in game design!

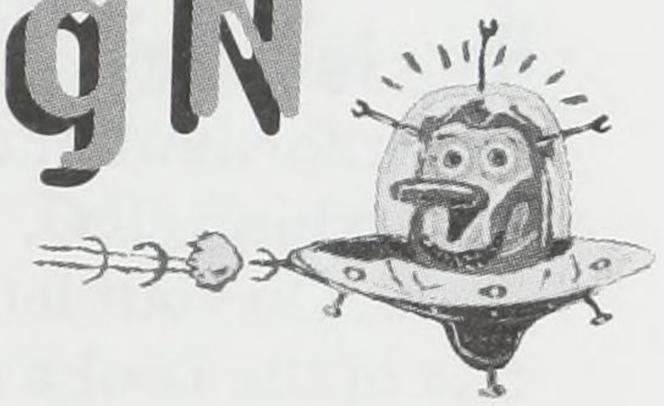
Arnie Katz
With Laurie Yates

Good luck
in your journey to the Underworld &
to the other games!
Bi-Rday!
computer
Happy B-D Guy

Brian
&
Andy
"1996

ISBN 1-55958-669-9

inSIdE eLEcTroNiC GaME deSigN



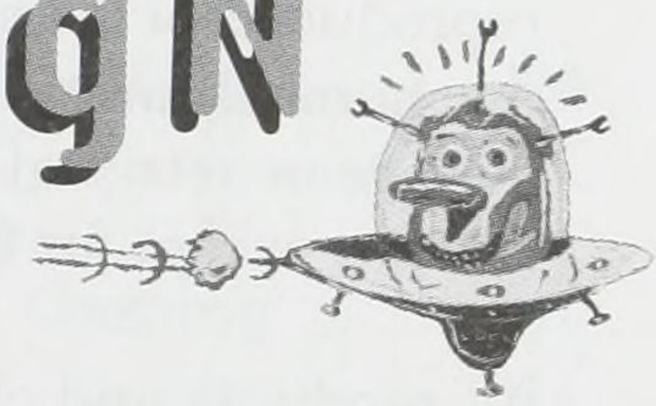
Now Available from Prima Publishing

Behind the Scenes at Sega
Klik & Play 1.0: The Official Strategy Guide
Make Your Point:
    The Complete Guide to Successful Business Presentations
    Using Today's Technology
The Making of Microsoft:
    How Bill Gates and His Team Created the World's
    Most Successful Software Company
The Software Developer's Complete Legal Companion
How to Sell & Promote Your Idea, Project, or Invention
How to Start Your Own Business on a Shoestring and
    Make up to $500,000 a Year

How to Order:

For information on quantity discounts contact the publisher: Prima Publishing, P.O. Box 1260BK, Rocklin, CA 95677-1260; (916) 632-4400. On your letterhead include information concerning the intended use of the books and the number of books you wish to purchase. For individual orders, turn to the back of the book for more information.

inside electronic game design



Arnie Katz
with Laurie Yates

PRIMA PUBLISHING

This book is dedicated, with great respect and affection, to the other founding partner of K & W, Inc.: Joyce Worley.

Secrets of the Games® and design is a registered trademark of Prima Publishing, a division of Prima Communications, Inc.

Prima™ and P™ are trademarks of Prima Publishing, a division of Prima Communications, Inc.
Prima Publishing™ is a trademark of Prima Communications, Inc.

© 1996 by Arnie Katz. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system without written permission from Prima Publishing, except for the inclusion of quotations in a review.

All products and characters mentioned in this book are trademarks of their respective companies.

Important:
Prima Publishing, Inc., has made every effort to determine that the information contained in this book is accurate. However, the publisher makes no warranty, either express or implied, as to the accuracy, effectiveness, or completeness of the material in this book; nor does the publisher assume liability for damages, either incidental or consequential, that may result from using the information in this book. The publisher cannot provide information regarding game play, hints and strategies, or problems with hardware or software. Questions should be directed to the support numbers provided by the game and device manufacturers in their documentation. Some game tricks require precise timing and may require repeated attempts before the desired result is achieved.

ISBN: 1-55958-669-9
Library of Congress Catalog Card Number: 94-67453
Printed in the United States of America
95 96 97 98 BB 10 9 8 7 6 5 4 3 2 1

Contents

Acknowledgments vii
Introduction ix

Part One: The History of Electronic Game Design 1
Chapter One: The Early Days 3
Chapter Two: The Golden Age 15
Chapter Three: The Great Crash 37
Chapter Four: The Renaissance 47

Part Two: The Theory of Electronic Game Design 59
Chapter Five: What Is a Game? 61
Chapter Six: The Aesthetics of Game Design 71
Chapter Seven: The Anatomy of Electronic Gaming 81
Chapter Eight: The Game Designer's Role 91

Part Three: From Dream to Disk 99
Chapter Nine: Sometimes a Great Notion 101
Chapter Ten: The Proposal 109
Chapter Eleven: The Preliminary Design 119
Chapter Twelve: Final Design Specifications & Beyond ... 133

Part Four: Meet the Designers 151
Mike Berlyn 153
Julie Carlson 155
Phil Goro 160
Rick "Coco" Gush 163
David Johnson 168
Peter Molyneux 172
Scott Orr 178
Jeff Peters 183
John Skruch 186

Jeff Tunnell 193
Will Wright 197
Rand and Robyn Miller 198
Patrick Charpenet and Hubert Chardot 202
Tim Schafer and Dave Grossman 205
Bill Crum 209
Rob Landeros 211
Jim Walls 213
Ted Barnett, Keith Kirby, and Rob Fulop 215

Part Five: Careers in Game Design 219
Chapter Thirteen: Preparing to Be a Game Designer 221
Chapter Fourteen: Breaking into Game Design 229
Chapter Fifteen: Designing the Future 235

Appendix A: A Sample Proposal 241
Appendix B: Contacts 249
Appendix C: School for Designers 263
Glossary 267
Index 271

Acknowledgments

This book could not have been created without the energy and dedication of many people at Katz Kunkel Worley, Inc. (as it was known), and Prima Communications. The author would like to take the opportunity to single out the following people for special mention:

Writer and critic Laurie Yates conducted and edited all the interviews with the designers in Part Four. Her enormous contribution provides a crucial element of *Inside Electronic Game Design,* the detailed studies of the lives of actual game-creators.

John Wesley Hardin furnished considerable aid and support during the preparation of this manuscript. His copyediting and proofreading whipped *Inside Electronic Game Design* into shape, and he contributed the appendix on schools for game designers.

Electronic Games' executive editor Bill Kunkel contributed heavily to the content and direction of this book. His keen insights into the subject, and his willingness to debate any point, solidified and strengthened the whole project.

Editorial Director of KKW, Inc., Book Division, Ed Dille helped design the project and wielded a sensitive, yet sharp, blue pencil through the entire gestation process.

I would also like to sing the praises of those fine folks at Prima, especially Roger Stewart, whose willingness to take a chance on such an unusual subject inspired me when the hours grew long and the monitor became hypnotic.

Brett Skogen, who took over the editorial reins in midstream, also merits several rousing cheers and accompanying pats on the back for putting the book back on track while dealing with a sometimes-wayward author.

Introduction

I wear many hats as president of Katz & Worley, Inc., which is an independent interactive electronic entertainment company located in fabulous Las Vegas, Nevada. Among other things, I'm involved with magazine editing and writing, book authorship, consulting, product evaluation, and game design.

Sometimes when I'm reciting this list to someone, especially when I'm trying to impress the person with what a Man of Stature I've become, I'll blurt out something like ". . . and I've helped design about two dozen computer, video, and multimedia games."

"Oh!" they say, suddenly impressed. "You program games!"

"No, I don't," I correct. "Programmers write the actual game code that executes when you boot the disk or cartridge."

"Oh!" they say, suddenly more bewildered than impressed. "Then you're the artist who draws what's on the screen."

"No, I don't do the graphics," I answer. "They hire people to draw and animate the games. I work with those people."

"Oh," they say, now more confused than anything else. "You run a team that makes the game."

"No," I reply. "The producer runs the team. As a designer, I work for the producer."

"Oh," they say, light suddenly dawning. "You come up with the ideas!"

"Sometimes," I respond. "A lot of times, the publisher supplies the idea in the form of a license."

I'll spare you the rest of this oft-repeated dialogue. It goes on for about fifteen more minutes, each of which feels like a subjective hour as I try to explain what exactly I do as a game designer to my increasingly suspicious and less-than-impressed audience.

Those conversations have had one good side effect: they've inspired this book, *Inside Electronic Game Design*. Now when game-lovers ask what, exactly, I do, I can give them a copy and avoid those endless explanations that wind up sounding a lot more like self-justifications and excuses.

Inside Electronic Game Design is expected to appeal to three types of readers: industry professionals (especially those who have been interviewed in Part Four), game lovers who always wanted to know how games were created, and anyone who desires a career designing electronic games. This book takes budding gaming designers on a tour of electronic game history, theory, and practice and outlines strategies for landing that first job.

Like most authors, I hope you read every word in this book, but I know that some people will read only the sections in which they think they need instruction.

Part One is a history of electronic game design in America. It chronicles the development of electronic gaming, from the first text adventures on mainframe computers to today's brave new world of multimedia. Games were not always designed as they are today, and the current position of "game designer" is only the latest step in an ever-evolving process.

Part Two presents the theoretical underpinnings of games. It dissects electronic games into their component parts and defines the principles that guide their construction. Gamers will increase their knowledge of the theories behind their favorite titles, while prospective game designers will learn what makes the heart of a game tick.

Part Three is the "how-to" section—it shows you how to turn an electronic game idea into an electronic game design. Starting from the raw concept, it leads you step-by-step through the proposal, the preliminary design, the final specifications document and revisions to the finished commercial product. Appendix A offers an example of a complete proposal for a hypothetical game.

Part Four introduces you to some of today's most-acclaimed game designers, who give you the inside scoop on what it's like to work in the industry. You'll learn who invented some of your favorite games and how they did it, what's most exhilarating and frustrating about creating games, and how these folks got into the industry in the first place. These professionals offer a wide variety of perspectives and personal histories—further

evidence that game designers are an idiosyncratic lot who come from all walks of life.

Finally, Part Five surveys careers in game design, how to break into the industry, and the trends that are likely to shape electronic gaming till the end of the century. Appendix B is a list of game companies and contacts, and Appendix C describes the only two North American universities offering training in electronic game design.

So whether you're curious about how games work, want a game design career, or you're super agent Barry Friedman checking out how much ink each of your clients got, I hope *Inside Electronic Game Design* fills the glaring knowledge gap. And remember: "The play's the thing."

PART ONE

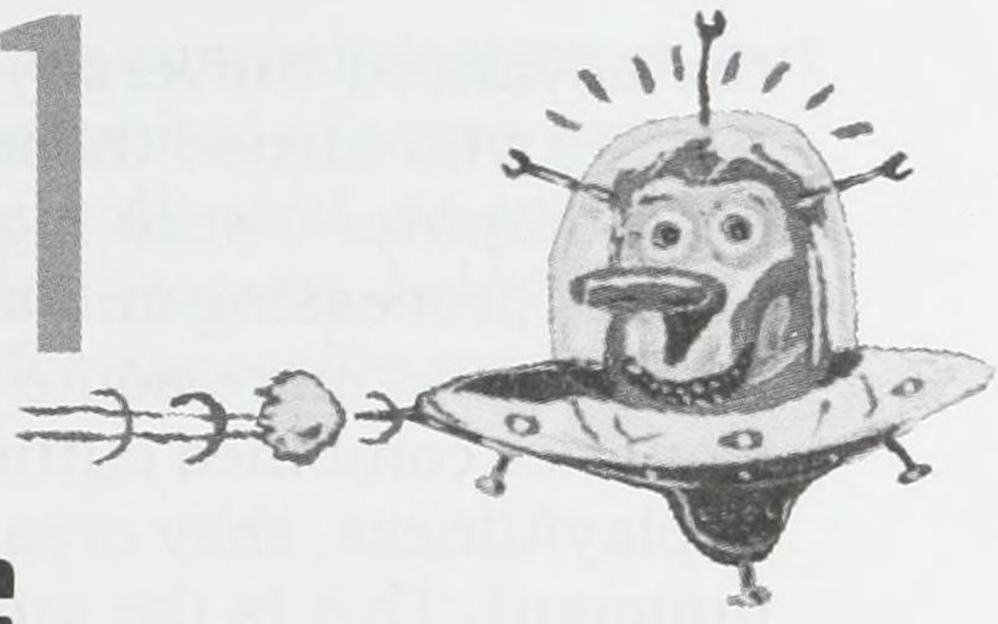
The History of Electronic Game Design

Learning the history of game design is part of the preparation for being a designer, but it also holds interest for anyone who loves and enjoys games. Game design as we know it did not spring to life full-blown overnight. It has developed gradually over three decades, the product of a diverse cast of characters and of relentless technology. It's a long, long way from the game design pioneers of the 1960s to today's software authors.

Today's games are many generations removed from the ones that delighted owners of the Odyssey[2] or Atari 2600. The technology is more powerful, and consumer expectations are correspondingly higher. What was good enough in 1978, or in 1988 for that matter, would lag behind today's cutting-edge efforts, in every respect, with the possible exception of the core play-mechanic.

The seismic changes in hardware technology have made game design a completely different job, requiring a different set of skills. If current software authors traveled back to the late 1970s, when game designers had to have deep technical knowledge of the hardware's capabilities, many would be unemployable as game designers. Today, a designer is the person who defines the content of an interactive entertainment product, an architect drawing the plans for the construction crew to implement. But the original "game designers," our forefathers, didn't even think of themselves as such. They were the computer hackers of a bygone era, when creating games was the seemingly frivolous hobby of otherwise serious computer programmers.

Chapter 1



The Early Days

The development of computers parallels the other electronic media, like radio and television. All started as grist for scientific inquiry, were initially dominated by serious applications, and finally became part of our mass media—in which entertainment and infotainment are often the prime attractions. Originally, government and university grants poured millions into the development of digital processing, never imagining that a multibillion-dollar interactive entertainment business would be a by-product.

During computing's stone age, from 1940 to 1954, programmers focused mostly on military applications. Turning a Nazi war dispatch into a readable communiqué may have gamelike aspects, but it was deadly serious for those antediluvian computer experts. No one had time to wonder if people would like to crack ciphers for fun in their living rooms (like *Codebreaker*, a cartridge marketed for the Atari 2600 video game machine in 1979). Computers, like tanks and airplanes, were part of the war machine.

Even after World War II, scientists conceived of electronic brains mostly as incredibly fast and flexible mathematical engines. They saw the wonder machines as a way to free people from abstruse and repetitive computations. A full-featured calculator would have satisfied many of their needs, had it existed.

The desire for entertainment cannot be denied. Fun often creeps into even the most serious work. The pioneering experimenters—

the advanced university science students and the kindred professors who turned those nascent electronic brains into the modern computer—were highly motivated and extremely dedicated to data processing and artificial intelligence, but the harder they worked, the more some of them needed to play. When a few visionaries combined cutting-edge technology with a maverick spirit of playfulness, they created the first interactive electronic entertainment. This is the story of how they did it.

In the Beginning . . .

The first electronic games inventors weren't designers at all! They were trailblazing programmers and computer scientists at universities like MIT and CalTech in the 1960s. They all worked as technical specialists and researchers. They wore pocket protectors, ate a lot of snack food, and were called "nerds" by those who didn't understand their enthusiasms. They were far outside the mass cultural mainstream, with almost no desire to understand that mainstream and certainly no interest in designing games that appealed to millions. Their research was conducted with little thought of practical applications, much less profit, and games were an incidental, if enjoyable, by-product of their fascination with digital technology.

Why did serious scientists spend time on games? Partly it was a product of their lifestyle as well as their alienation from popular culture. They worked in unorthodox ways and almost never had set working hours. Marathon hacking sessions—writing computer code "on the fly" without a formal plan or outline for several days at a time—were not only expected but became bragging points.

Most subscribed to the philosophy of hacking. The hacker ethic brushes aside convention, and a few minor laws, in the pursuit of knowledge and the satisfaction of curiosity. These computer scientists did a lot of serious coding, but their maverick natures and near-addiction to the hardware led them to toy with more frivolous uses. Also, since they drove themselves relentlessly, creating games became a way to blow off steam, even though they sometimes had to come up with a good line of double-talk to justify the time they spent on such noncritical projects. Sometimes, they even found a way to get a few bucks for the effort.

Hacking is not exactly unknown among current designers, but it is becoming rarer. Today, it's far more typical of game *programmers*. (We designers are supremely well-adjusted, at ease in every social situation, suave men and chic women, beloved by all who know us. We'd be perfect, except for excessive modesty.)

The First Electronic Game

But who actually invented electronic games? As is the case with the National Pastime, there is confusion in the popular mind about who invented the first interactive electronic entertainment product targeted at the home market. Despite what you may hear and read in the mass media, the answer is *not* Nolan Bushnell. Atari's founder is a seminal figure, but neither he nor the inventor of the first video game, Ralph H. Baer, are legitimate contenders for this distinction.

MIT hacker Alan Kotok produced an early computer chess program, but there's no evidence that this was the first attempt, or even the first successful one. We'll probably never know who was the first to encode Tic-Tac-Toe or Hangman in the wee hours of a now-forgotten night, never thinking that such "goofing off" might one day seem significant.

If baseball can claim Abner Doubleday as the inventor of the National Pastime, when all evidence points to Alexander Cartwright as the codifier of baseball's modern rules, then electronic gaming ought to be able to agree on a symbolic creator as well.

My nominee for the first *original* computer game is *Space War*. Considering all the momentous discoveries that MIT made in the late 1960s—including nailing down the theory and practice of computer operation—the hackers can be forgiven for hardly noticing what they'd done. *Space War* wasn't even created to be a game in and of itself, but as a means to demonstrate the graphic capabilities of the latest technology. That's how it snuck up on the MIT computer community.

The arrival at MIT of the DEC PDP-1 (initialese for Program Data Processing) computer in 1961 set events in motion. With more capabilities than the TX-0 ensconced in an adjoining lab, its easier-to-program CRT cried out for some kind of eye-popping demonstration. When more serious efforts failed to generate the sizzle for the necessary demo, Steve "Slug" Russell

came forward with an idea: design a game that gives the computer a workout.

A longtime fan of pulp science fiction, especially the universe-spanning novels of E. E. "Doc" Smith, Russell created a rudimentary space war game with a lot of help from people moonlighting from more important research. *Space War* went through many intermediate stages as various people added their ideas, and the final product exceeded what any one of them could have accomplished alone.

Russell was a programmer first and foremost, not a game designer. He had to be a code cruncher, since only the most computer-literate people could make the PDP-1 do anything meaningful. At that time, code had to be punched into a tape, which only authorized personnel could insert into the machine.

Alan Kotok, still celebrated today as a seminal hacker, helped Russell past the first big hurdle, the need for a sine/cosine generator; once Russell had that he could try to put something on the screen. By winter 1961, he could govern the speed and direction of an on-screen dot by flipping control-panel toggles. The "ship" could turn clockwise or counterclockwise, accelerate, and fire. (Similar play-mechanics are found in such later games as *Asteroids* and *Astro Chase*.)

That may not sound like much, but it was the equivalent of climbing Mount Everest. Until that dot started zipping around the screen, no basis existed for an electronic action game.

Picking up momentum, Russell created two rockets that could be moved like the dot, produced a subroutine to fire a dot from one to the other, and then created a search routine that let the machine determine if the missile occupied the same screen position as the target ship at any time during its flight. If it detected that event, Russell's program wiped the ship image off the screen and replaced it with a spray of dots. In a matter of a few months, Russell had developed the essentials of an electronic target contest, a head-to-head shooter: objects, missiles, and a collision detector.

When Russell threw *Space War* open for comment to the rest of the local computer community, he scored another historic first: the first electronic game design fight. In his basic design of the shooting program, Russell worked a random element into the flight paths of the missiles on the grounds that such devices

aren't unfailingly accurate. His friends overruled this, collectively becoming the first electronic game producer (the person who is the final arbiter of the intellectual property the team creates). They substituted a simple routine that allowed players much greater accuracy.

Other changes, ostensibly more revolutionary, flowed into the ever-more-sophisticated game. Peter Samson designed a mappable, astronomically accurate starry background. Dave Edwards felt that *Space War* didn't have enough strategy so he put a sun in the center of the screen. Skilled pilots could use the stellar gravity well to augment their ship's speed, though spiraling into the star incinerated the vessel. Shag Garetz borrowed a familiar science fiction concept and gave players a "panic button" that caused the ship to jump to another, random screen location, one hopefully out of immediate danger.

Space War gained great popularity among the computer insiders at MIT, but there was no question of commercial exploitation. After all, who else had a PDP-1 to run it? *Space War* began and ended as an academic experiment.

The project had one almost-accidental consequence of great import to electronic gaming. One of the lesser lights of the MIT scene came away from the experience with a profound belief in the technology's entertainment potential: Nolan Bushnell, the founder of Atari.

It is no insult to Bushnell to say that he didn't rank with Kotok or Russell as a computer scientist or programmer. He was, instead, a visionary who also had the practical bent necessary to take electronic games out of the lab and put them into the marketplace.

Like many other visionaries, success came to Nolan Bushnell only after repeated failures that would have demoralized someone of lesser stature. He believed *Space War,* or something like it, would be a great alternative to pinball and similar amusements in arcades and bars, but he couldn't seem to find the successful formula. Again and again, he designed games that he hoped would capture his target audience. Always the machine broke down, or coin-op pros dismissed his proposals as too complex to appeal to the typical pinball wizard. A few *Computer Space* prototypes, which Bushnell based on *Space War*, can still be found in collections and a few arcades, but it never went into mass distribution.

Reports vary, but Bushnell may have tried as many as *nine* versions of *Space War* before admitting that he couldn't turn the design into something one of the coin-op companies would buy. It's ironic, in view of the games that have followed, that complexity proved the chief stumbling block. The Bally Manufacturing Company evidenced interest in the newfangled contraptions, but Bally execs didn't think the typical arcader would spend money on anything with so many rules and play-elements. Elegant simplicity characterized most of the more popular pinball machine designs, and the coin-op king wanted to see the same trait in this daring new category.

Nolan Bushnell lived through electronic game design hell. Instead of creating something, he found himself demolishing what he had, like some sort of retro-designer. Each successive version of the *Space War* coin-op eliminated aspects of the previous versions. Instead of elaborating on the basic structure, he was forced to simplify even further in order to get a toehold in the coin-op world.

Everything changed in the early seventies when Bushnell accepted an invitation to view a demonstration of prototype video technology. The demonstration was of a video tennis contest on a TV screen, and when Bushnell saw it he knew he'd found the perfect game design. Reluctantly, he gave up on *Computer Space*.

Instead, Bushnell constructed an electronic coin-op machine simple enough to attract casual play and sturdy enough to hold up under player abuse—the kind of coin-operated game machine found in commercial video arcades today—and he called it *Pong*. When he put it into a neighborhood bar, patrons jammed so many coins into the slot that it temporarily stopped working. That's the kind of problem any manufacturer wants to have. It wasn't long before *Pong* machines sprouted up across the country and Bushnell had a growing company he named Atari.

The Godfather of Video Gaming

But though Bushnell is credited with creating the first video coin-op, he didn't invent the video tennis contest it played. Though it isn't generally known, Atari paid royalties on the design of *Pong* to the New Hampshire outfit, Sanders Associates, that had sponsored the demonstration.

New Hampshire doesn't rank with Silicon Valley as an electronic beehive, but Sanders Associates was well known for sophisticated military electronics. Ralph H. Baer, the designer of *Pong*, joined the company in 1959. Baer quickly rose to division manager at Sanders, which was the state's largest employer.

Idle speculation about additional uses for TV sets led Baer to the concept of electronic games. Finally, in 1967, he helped organize a formal project, which consisted of two assistants who were kept locked in a small room to promote secrecy.

The team built four or five prototype game machines. While testing these devices, Baer and his hardware engineer colleagues almost unconsciously slipped into game software design since they needed to have something to show people. "We had *Ping-Pong*, rifles to shoot at the screen, board games, and sports," Baer recalls.

Unlike modern video game consoles, these were hardwired machines, meaning that each one could only execute the instructions for a single game. The Sanders Associates crew were primarily hardware engineers, but they also displayed a flair for concocting games simple enough for the crude equipment to present.

How crude? The equipment was all analog, since microprocessors hadn't been invented yet. Technology had not yet advanced beyond transistors, which had replaced the vacuum tube used in early computers like Whirlwind, the code-breaking computer.

Unfamiliar with the consumer market, Sanders Associates shopped the video game idea around the business world. In view of today's telecommunications explosion and the debut of the Sega Channel, it's interesting that Sanders Associates first offered its brainchild to cable operators. Far ahead of his time, Baer wanted to download background graphics for video games, called playfields, to subscribers over the wire.

Though it's hard to believe today, the cable companies passed on the deal. The clarity of hindsight reveals this as a colossal blunder, but at the time the infant cable business was still caught up in birth trauma. Even giants like TelePrompTer didn't have the money to fund the project. Survival, not diversification, was the watchword in cableland.

Baer soon set his sights on another branch of the electronics business. "In 1970, we began parading it past the TV set manu-

facturers," Baer says. "We showed it to everyone who made televisions in this country, which at that time was still quite a few companies, including RCA, Sylvania, and General Electric."

RCA entered negotiations, but talks broke off without an agreement. Then Magnavox, a division of North American Philips, asked for a demonstration. By this time, the Sanders crew had a lot to show. Since Baer was a hardware designer first, his initial games were quite simple and stressed the use of the joystick, gun, and other clever controllers. This was a significant development in gaming history, the first time designers tailored play-mechanics to fit special controllers.

The most innovative Sanders device was a harbinger of virtual reality golf simulations to come. It used a golf ball mounted on the end of a joystick. Striking the ball caused corresponding movement on the screen. They couldn't do hyper-realistic sound and graphics, but the play-action was authentic.

"We also demonstrated *Ping-Pong*," Baer reports. "Unlike *Pong*, our paddles could move all over the screen, and you could use 'English.'" Redirecting the flight of the ball after it was hit became the main tactic.

Perhaps Bushnell, or possibly his benevolent secret advisors at Bally in the coin-op industry, felt that unrestricted paddle movement was too much of a good thing for potential players. They decided to restrict paddle movements to the horizontal plane. It made the game quicker to learn and easier for players to follow during volleys. The limited interaction of pinball was the arcade norm, and Bushnell may have worried that allowing too much freedom of movement might confuse and alienate players weaned on flipper tables.

When they cut away all the complications in *Ping-Pong*, the result was *Pong*. Bushnell had his ideal design, a contest for one or two players that anyone could learn in thirty seconds and play for hours. *Pong* exudes the elegant simplicity that has characterized megahot games from *Pac-Man* to *Wolfenstein 3D*.

The first commercially successful video game, *Pong* teaches a design lesson that is still valid today. Heaping rules and systems on a basic design doesn't necessarily improve the game. More often, well-defined goals and a streamlined play-mechanic are what win the public's love.

While Bushnell and his colleagues readied their assault on America's pocket change, things weren't going as smoothly for Baer, Sanders Associates, and the effort to link games to television.

It wasn't exactly love at first sight when Baer and his allies made their presentation to Magnavox. It came close to a "no sale." Had it not been for one North American Philips honcho, a now-forgotten vice president, the company would have passed on video games. This far-seeing executive pushed through a small project to build a few machines in 1971. The results of the pilot project made Magnavox happy enough that it introduced the Odyssey (Model I, TL 200).

Caught in the Switches

The Odyssey, the first programmable cartridge video game unit, brought electronic gaming into American homes, but it had significant limitations. Like almost every game console since, the Odyssey was an underpowered machine without many features. The state of chip technology, and its cost, led Baer to a lot of tough decisions. The Odyssey was consequently saddled with very limited capabilities that circumscribed cartridge design throughout the system's several-year lifespan.

For example, the Odyssey didn't even have enough memory to put the playfield on the cartridge; every Odyssey game included a plastic overlay that adhered to the TV screen during play. The germ of the idea may have come from a 1950s kiddie show, "Winky Dink," in which the viewing audience drew on a TV screen overlay in predesignated spots, allowing apparent interaction with the program's animated sequences.

Whatever the source of its inspiration, the Odyssey offered little scope for designer wannabes: A new overlay and a cartridge that reset parameters for on-screen cursor movements turned an auto race into a hockey match.

This was not the age of the designer, but of the marketing maven. Game manufacturers needed strong, simple concepts, since the Odyssey could present nothing more than a streamlined and abstract reflection of reality. Once the company had come up with some vague generality that sounded like it might sell—such as an auto race—it fell to the hardware engineers and programmers to turn the idea into something that the Odyssey could run.

Electronic gaming slid through the early and mid-1970s in this fashion. Hardwired video game units and stand-alone devices (using LED and then LCD screens) flooded the market, but the design component was too primitive to require specialized expertise.

Most who got to design games did their work "unplugged." They designed board games, including military simulations and fantasy role-playing games for companies like Simulations Publications (SPI), Avalon Hill Game Company, and Game Designers Workshop. And in their off-hours, they haunted the arcades, playing *Pong* and dreaming of better days.

The "Nice Try" Programmables

To the untrained eye, video game technology inched ahead at a glacial pace during the first half of the seventies. Behind the scenes, developments unfolded with the suddenness of a midwinter blizzard. Development of more powerful microprocessors and more efficient memory storage units gave pioneering electronic gamemakers the tools they needed.

Everyone in the industry could see that hardwired games were a dead end. A cartridge system boasted more flexibility at a lower retail price. Consumers might be willing to buy one *Pong* machine, but they would never buy a new machine every time they wanted a different game. The beauty of a cartridge system is that consumers buy the hardware once, instead of purchasing it anew for each game. Though the Odyssey's programmability was a wonderful feature, the unit itself didn't have the graphics or data-crunching power to do much with it. In this one respect did single console games like *Pong* have an advantage. But hardware engineers believed they knew the answer: Put the operating system and other permanent code into the console and put only the specific instructions for the game in the plug-in cartridge.

RCA's programmable console reached market first, around 1975, but it made no dent. Black-and-white graphics and a collection of games little better than *Pong* kept the RCA Studio from developing a following.

Fairchild's Channel F, which made its debut in 1976, had improved color graphics and smaller controllers, but their games were also not very good. The company wasn't very experienced

with this sort of entertainment, so Channel F, too, couldn't penetrate the market to any great extent.

The Bally Professional Arcade debuted in mid-1977. It could have changed electronic game design history, but it didn't. Its high price ($299 retail) and daunting promotional rhetoric prevented mass market sales. Many saw it as a junior computer, not a high-voltage video game machine, and the public wasn't yet ready for that leap.

This machine, conceived by the same company that earlier disdained video coin-ops, was utterly different than the ones that ultimately succeeded with the public: the Atari Video Computer System (VCS) and Magnavox's Odyssey[2]. The VCS, the most popular machine, was a black box like the Nintendo Entertainment System or Sega Genesis—with the innards of its console off-limits to the user—but the Bally system let the average person actually program his or her own video game designs. Its console incorporated a calculator-style keyboard, 8K of memory, and ports for up to four controllers. These were a story in themselves, combining joystick and paddle functions in one compact unit.

Bally's pride and joy was a cartridge that taught owners how to program in BASIC, a "plain English" coding system popular in the late 1970s and 1980s. Programs could be stored on tape cassettes and transferred to the Pro Arcade.

The system's documentation and advertising spoke confidently of how Professional Arcade owners would advance from playing prepared software to creating their own. It was no idle dream. User groups formed around the unit generated many programs, including games, until Bally stopped supporting its stillborn progeny. (Astrovision bought the project and relaunched it in 1981, but without success.)

The Bally Professional Arcade had four resident software titles: a calculator, a drawing program, and two games. *Gunfight* made excellent use of the unit's controllers. The one- or two-player side-perspective shoot-out armed buckaroos with six shots each. The gamer moved the gunfighter in a limited area with the stick, aimed with the paddle, and fired by pressing a button.

Checkmate, the other game, was a one- to four-player kinetic strategy contest with the same play-mechanic as Atari's later *Surround*. The player guides a constantly elongating line so that the head didn't intersect any other line, on-screen obstacle, or the playfield boundary.

These games set the style for the system's software library: streamlined, action-oriented play-mechanics; a dash of strategy; and colorful but simple graphics with no playfield background detail. In short, these games were fun to play without being so complex that smart Bally Professional Arcade users couldn't imagine programming something as good or better.

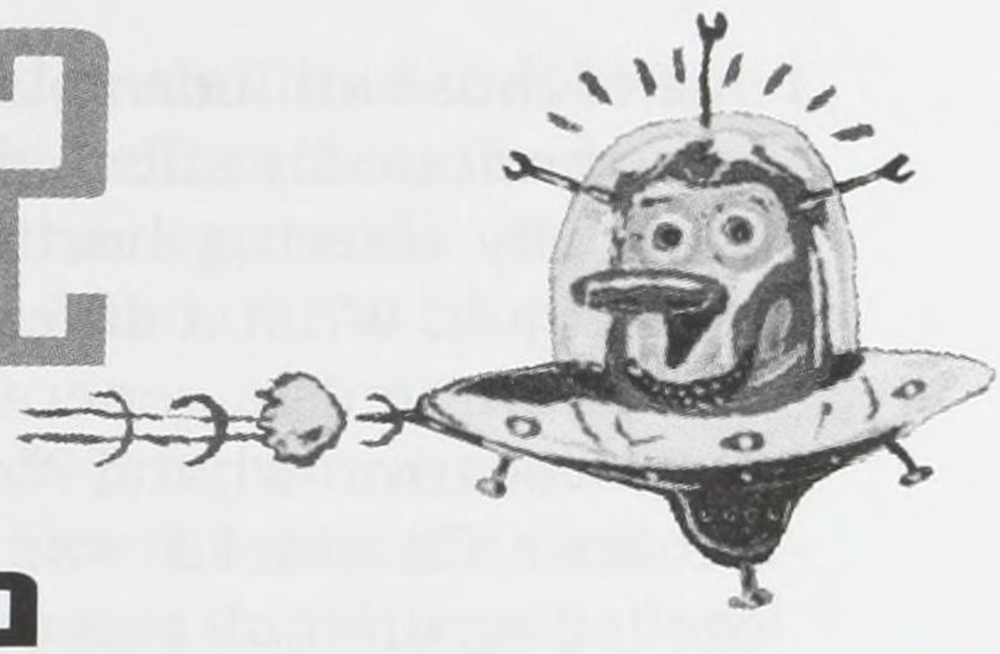
The Stage Is Lit

Virtually all of the systems described so far must be classified as partial successes. They fired the public's imagination and whet consumers' appetites for this type of experience. In that sense, they were the necessary precursor to the world of interactive electronic entertainment as it is today.

Of all the aspects of interactive electronic entertainment discussed so far, only coin-ops translated the glamorous, cutting-edge image into a sizable on-going market. The pay-for-play machines became an integral part of amusement centers in the 1970s, but nothing comparable happened in the home sphere. A few electronic toys, like the various home *Pong* units and Mattel's *Electronic Football* sold well, but no attempt to entertain with computer technology achieved market penetration comparable to videocassette recorders in the 1980s, let alone stereo and television.

Yet, like a Broadway play just before the curtain rises, everything was in readiness. The technology had become more powerful, interactive electronic entertainment had acquired a favorable image, and manufacturing efficiencies had lowered retail prices. Game-lovers could feel, with some confidence, that it was only a question of time until reality caught up with their dreams.

Chapter 2



The Golden Age

Some historians contend that innovations such as the steam engine occur when it's "steam engine time." When society is ready for something, it happens. That's why, say proponents, seemingly revolutionary inventions like radio are often discovered virtually simultaneously by several individuals.

For instance, if the Wright Brothers hadn't invented the airplane, someone else would have. The technology was available, and society had potential uses for an airplane. Given those factors, the airplane was inevitable, and the only question was who would have the honor of actually making the breakthrough.

If the theory is valid, "electronic gaming's time" arrived in 1978, ushering in a Golden Age of Video Gaming that lasted until 1984. Rapid changes and a generation of new products made electronic gaming an identifiable hobby and opened the door for careers in electronic game design.

The Road Not Taken

The Atari 2600 and the Magnavox Odyssey[2] rocked home entertainment in 1978. Their instant retail success launched video gaming's "golden age" and did much to shape the game design environment for years to come.

Could the Bally Professional Arcade, or even the Fairchild Channel F, have cracked through eventually? It's not likely, considering their deficiencies. Yet it could have happened. And had

one of those stillborn platforms overcome its handicaps, it would have profoundly affected the role of the game designer, since dramatically altering hardware capabilities radically changes the design job. What a difference that might have made to electronic entertainment in general, and game designing in particular.

Provocative "what if?" scenarios about the Channel F are scarce because Channel F was just too primitive. Had it somehow succeeded against all reason, Channel F's shortcomings might have blunted the first video game boom and set the technological timetable back a half decade or so.

The Bally Arcade is another, more tantalizing question. Better marketing and merchandising, plus a lower price, could have turned this gaming computer into a hit.

If the Bally Arcade had caught the public's interest, the US game business might be more like its British counterpart. In the United Kingdom, small computers filled the niche occupied by cartridge consoles in the United States. Instead of closed boxes like the Atari 2600, young Britons bought fully programmable computers. While America's youth perfected *Pac-Man* strategies, British youngsters dabbled in programming their tiny machines. That's partly why there are more game software programmers per capita in the United Kingdom than in the United States.

Bally actively encouraged customers to try coding their own stuff. Widespread popularity for the Professional Arcade could have rewritten the subsequent course of game design history. It might have ushered in an era of do-it-yourself video game design. The shareware boom of the late 1980s could have started a decade earlier.

When Cartridges Were King

Neither the Professional Arcade nor Channel F set enough pulses racing, and just as these hardwired devices started to fade, North American Philips (NAP) and Atari introduced their programmables. The architecture of these cartridge systems, as well as all subsequent cartridge players, shaped the design function.

The Odyssey^2 and the Atari VCS, soon renamed the Atari 2600, were generally similar but radically different in their particulars. The Odyssey^2 had a flat, or monoplanar, keyboard and a chip set that favored plain backgrounds and intricate foreground images.

The 2600 was a black box with a cartridge slot, and it sacrificed a little detail in the sprites, or foreground characters, in favor of enhanced capability to create playfield backgrounds. Atari also boasted faster execution speed, which turned out to be crucial in presenting arcade-style action games.

Both Atari and NAP employed designer-programmers to create cartridges. For game design, this was like the day the first amphibian crawled from the primordial seas onto prehistoric land. These designer-programmers were not as skilled as designers today, but then again the technology wasn't as sophisticated either. The first machines had a chronic lack of built-in, "hard-coded" memory ("read-only memory," or ROM). Hardware engineers, often unfamiliar with the requirements for games, invariably skimped on memory, sound generation, and sprites. This is a mistake that is still too often made. Just as soon as a hardware inventor proclaims, full of self-congratulation, "No one will ever use *that* much memory," events conspire to make the statement look foolish. The future may unravel this sweeping axiom, but no electronic game-player produced so far has ever had as much memory as it could use.

Atari designed the 2600 to run 2K cartridges. Within a year, designers delivered 4K games that involved subtle manipulation of the hardware. The public noted the improvement over first-generation cartridges and clamored for further advances.

Not too long after came the answer to their prayers: bank-switching. The designer-programmers took apart the 2600 like a wooden cube puzzle. They discovered that it was possible to use the 2600's two memory banks (physically separate sets of memory chips) in rapid alternation to execute the program code. Bank-switching allowed the machine to run 8K cartridges.

It took top programming ability to tailor games to the 2600's eccentric configuration, so Atari tended to let its programmers double as designers. No one got royalties on their work, so doubling up saved a designer's salary as well.

The Odyssey^2, handicapped by its sophistication, couldn't be manipulated as cleverly. Its complex microprocessor was harder to fool with simple programming routines. NAP never discovered bank-switching or its equivalent. This forced an emphasis on games that offered little more than good play-action in a sparse audiovisual environment. It proved convenient to hire

programmers and let them provide cartridge content, with a little guidance from the marketing department and concept gurus like Steve Lehner.

However, for all but a small handful of Atari and NAP employees, the whole subject of game design was academic. Most of the work was done in-house or with a very small number of independent contractors. Video gaming was a closed shop with few entry-level positions. The people who invented the games didn't share in the profits, and there was no wider video game industry to speak of. Mattel and Coleco eventually got into programmables, but their activities centered on hand-held electronic games until the end of the seventies.

The Third (Party) Man

If video game design has a patron saint, it must be James Levy. The record business veteran did something that aided the cause of game design more than anything else: he created the first third-party video game company, which publishes software titles independently for already established game systems. Soon other independent software publishers followed, and as bidding for design services heated up, pay and opportunity for designers exploded.

In 1979, Levy created Activision with four top Atari designer-programmers—David Crane, Alan Miller, Larry Kaplan, and Robert Whitehead. The subsequent addition of Crane protégé Steve Cartwright, a facile designer-programmer, further solidified Activision's niche as a progressive video game design/development house.

Atari was not amused. Even ignoring the brain-drain caused by the independent publisher, Atari saw Activision as a poacher. Atari wanted to make all the money from its 2600, including all the software. The company's management fretted about how Activision would divide the market and siphon off money otherwise earmarked for Sunnyvale, CA.

Atari pursued a vicious campaign against Activision. They wanted to kill it dead and discourage others from following Levy's declaration of independence. Tactics ranged from threats of legal action to rumors that Activision cartridges damaged 2600 consoles. Atari even strong-armed journalists with hints

that coverage of Activision titles could endanger the future supply of Atari carts.

The bullying proved futile. Nearly everyone else in the industry immediately recognized that Activision was a huge, positive step forward for video gaming. Neutral observers saw what eluded Atari's braintrust, that Activision's products and promotions would raise consumer interest and expand the entire market.

That's exactly what happened. The Atari 2600 had only a slim lead over the Odyssey^2 when the third-party publishers jumped into the business. NAP was so busy congratulating itself on keeping tight control of its cartridge operation that it barely noticed how the rapid expansion of the 2600 software library blew open the market. By the time *Electronic Games*, the first newsstand magazine devoted to electronic entertainment, began reporting the field in late 1981, the 2600 had pulled away from the Odyssey^2 and become the dominant system. The 2600 went on to sell more than 26 million machines, while the Odyssey^2 stalled short of the 2 million mark.

Apollo, US Games, Imagic, and many other cartridge publishers plunged in as soon as they saw that Atari couldn't put Activision out of business and that Levy's all-star team was a moneymaker.

David Crane became the first celebrity designer-programmer. Activision merchandised its creative stars the same way a book publisher touts its best-selling authors. This approach, never before used in the video game field, propelled the company to a unique status among the expanding ranks of third-party cartridge publishers. Making sure that consumers knew who had designed each title helped establish Activision's "house style": addictive play-action with cartoon-style graphics.

Crane's *Freeway*, in which a chicken dodges traffic to cross a multilane road (a clever twist on the coin-op *Frogger*), gave Activision an early hit. A zany, if mildly scary incident at the 1980 Consumer Electronics Show in Chicago inspired *Freeway*. When Crane and some Activision colleagues left the convention center from the wrong exit, they had to cross Lakeshore Drive to get where they wanted to go. As they threaded their way through the onrushing lanes of traffic, someone remarked that this was just like a video game. Within a year, it *was* a video game.

Larry Kaplan, primarily a hardware designer who eventually left Activision to pursue that interest, followed Crane's lead with wonderful results, the classic *Kaboom*. The drawing and animation of the masked bomb-dropper thrilled consumers, as did the play-action. Like most Activision entries, *Kaboom* didn't try for depth and complexity. Kaplan devoted most of the memory to the visuals and built the design around a classically simple, yet addictive, play-mechanic lifted from the *Avalanche* coin-op.

Another Activision founder, Alan Miller, crafted many enduring design innovations before he started his own company, Accolade. His most lasting contribution is the trapezoidal playfield. Premiered in *Basketball* (Atari/2600) and perfected in *Tennis* (Activision/2600), its illusion of depth made it possible to express three-dimensional movement in a two-dimensional medium.

"Borrowing" the ideas of established arcade games was much more prevalent than inventing new games for cartridges. Crane, Kaplan, and Miller at least borrowed imaginatively, but many video games were transparent swipes, and the early computer game "designers" stole with an unparalleled audaciousness. Knockoffs of such popular coin-op hits as *Asteroids*, *Galaxian*, *Pac-Man*, and *Space Invaders* were endemic on disk. Once a publisher picked a concept, the next step was often to dispatch the programmer to the nearest arcade to scout for new ideas.

During the Golden Age, this was standard practice. Nearly everyone did it to some degree, and it wasn't seen as dishonorable. Designers were usually quick to credit the sources for their games, even if companies preferred not to dwell on it. But eventually—call it karma—the game companies that blithely took the work of others eventually suffered from their own lax attitudes toward piracy when rivals cribbed their best ideas and programming routines.

Playing on the Mainframe

Computer game design didn't get much beyond *Space War* until the late 1960s. There wasn't much incentive, since home computing remained a rarefied, do-it-yourself hobby until the late 1970s, when the Apple II and, to a lesser degree, the Commodore PET replaced the TRS-80 and the expert kits as the leading home computers.

However, there *were* mainframes. Corporations began aggressively installing them in earnest in the 1960s, which greatly increased the number of people with understanding of, and access to, computer paraphernalia. Mainframes were "serious stuff." Such vast computational power (now easily eclipsed by modern microcomputers) must only be used for Good lest its power be subverted to Evil Ends.

In the late 1960s, William Crowther and Don Wood wrote the first permutation of a text-based game called *Adventure*. Crowther allegedly wanted to concoct a mind-stretching entertainment for his kids, but it wasn't long before the program infiltrated mainframes around the country. Once-dedicated computer operators took sunrise trains, brown-bagged lunch, and stayed long after quitting time to wrestle with the logic puzzles encountered in *Colossal Cave*, another early game.

Even a compu-illiterate could instantly grasp *Adventure*'s essentials. The program prints a paragraph on the screen that outlines the situation. The narrative always mentions several objects, one or more of which the gamer can use to resolve the challenge. The player types a command in a strict noun-verb format to tell the program what he or she wants to do next. The program's parser recognizes key words in the user's input and emits the proper result paragraph. (An important nongaming application of this principle was *Eliza*, a late 1970s computer-shrink who presided over interactive therapy sessions.)

If the player types a noun-verb combo that the program decides is the right answer, it moves the adventure along with a new text block that presents the next puzzle. If the computer recognizes the player's words as a wrong answer, it runs an adverse text block. Players, especially inexperienced ones, often peppered *Adventure* with responses that exceeded the parser's capability. This stalemate triggered a call for a retry. The gap between what the user wanted to do and what the program could understand made *Adventure* frustrating for many. Still, you *could* eventually find the right word if you kept typing, and *Adventure* was so enthralling that even this inconvenience didn't kill the enthusiasm.

Adventure probably caused more than one cost estimate to come in a day late, but its importance to game design is incalculable. Anyone who could program even a little could encode a

game like *Adventure*. It required very few instructions, none beyond the competence of anyone who has read a beginner's book of BASIC programming. Sound and graphics played no role, so *Adventure* required no audiovisual programming skills. Those who didn't want to get even that involved in programming could simply replace the game's existing text blocks with their own puzzles.

The Disk Dreamers

The computer is unique among consumer electronics devices in its ability to generate copies within itself. A single computer can churn out copies of a program twenty-four hours a day, with no second machine required. Blank disks are cheap, especially if bought in bulk, and a Ziplock sandwich bag provides adequate protection. With the addition of a cover sheet, duplicated at the local copy shop, this was the standard computer game packaging circa 1980.

The cost of entry into the recreation software market was minimal. This encouraged entrepreneurs who either designed a pleasing game or knew where to find one to go into business for themselves. To call these tiny companies shoestring operations overstates their size. They frequently boiled down to a single designer-programmer and whomever he or she could con into helping with the boring chores of stuffing and mailing.

Among those who enjoyed *Adventure* and its imitators on the mainframe was Scott Adams. Except for his Florida locale, so far removed from Silicon Valley, Adams typified the pioneers who were guided by their individual visions. Adams started Adventure International in 1979 to bring text adventures to home computing. *Adventureland*, his homage to Crowther and Wood, and nine more major releases followed in fairly quick succession on tape cassette.

The Adventure International line did not improve much on the basic structure of *Adventure*. The line succeeded because the puzzles remained a step ahead of the players, and there was nothing nearly as engrossing available for those expensive desktop systems. (Adams re-released his microcomputer adventures on 5-1/4-inch floppies and then redid the entire line with a large, coloring book-style illustration for each major scene.)

Many of today's large computer software publishers trace their origin to the 1979-81 period—and to the work of one or more remarkable designers. Ken Williams started On-Line Systems, now Sierra On-Line, with *Mystery Funhouse*, the first of Roberta Williams' Hi-Res Adventures. The most successful title in the series, *Wizard & the Princess*, followed within the year. It was a text adventure with accompanying on-screen line illustrations that raised adventures to the next level of sophistication. Between the release of these two titles, Sierra acquired *Softporn Adventure*, coauthored by Ken Williams himself, a text adventure that is the acknowledged precursor to Al Lowe's *Leisure Suit Larry*.

Jon Freeman and Jim Connelley, who met while playing non-electronic role-playing games (RPGs), started Automated Simulations to distribute their *Starfleet Orion* in 1980. The company, which later became Epyx, listed the "Apshai" role-playing games and *Star Warrior* among its earliest products. Bill Stealey launched MicroProse, and Jerry Jewell brought the action contests of Nasir Gebelli (*Space Eggs*, *Autobahn*, *Pulsar II*) to the home computer through Sirius Software. The Carlston siblings issued the *Pac-Man* clone *Snoggle* and then a lot more original work under the Broderbund label.

The Flowering of the Computer Adventure

Adding illustrations to adventures, an ostensibly minor enhancement, revolutionized adventure/RPG design. The stories now had concrete settings, and the plot had to progress through a series or network of locations, depending on the degree of linearity, and this changed the way adventure creators visualized their work.

Until Sierra On-Line produced the first illustrated adventure, titles in this genre were just a series of puzzles thrown at the player one at a time in lockstep order. An introduction might say that your character is trudging through a series of caves or the corridors of a labyrinth, but the games conveyed little sense of place.

How could they? Illustrated adventures raised the artistic level of this category. Text adventures avoided lengthy descriptive passages since they could muddy the puzzle. Discursive and dramatic prose encouraged the user to enter a lot of words the parser couldn't translate, shooting aggravation sky high.

Pictures identified a location quickly and unmistakably without introducing red herrings into the prose.

Now designers could think about computer adventures the way that designers put together scenarios for nonelectronic role-playing games like TSR's *Dungeons & Dragons.* A structure based on geographical relationships, the passage of time, and cinematic scenes replaced the puzzle-to-puzzle link.

Text stayed simple in the first illustrated adventures, but designers discovered that players liked having several interactions at each location. "If, then" program statements permitted some conversation with nonplayer characters, which led the category away from its former dependency on conundrums.

Marc Blank and his Infocom gang had a different idea. They hated the graphics adventures. If the text could be elaborated, they claimed, the mind would conjure much more convincing images than the crude drawings incorporated into the new illustrated adventures. Instead, Infocom developed a stronger parser. This program had a much wider vocabulary, one rich in synonyms, that reduced the guessing. The Infocom parser also allowed use of adjectives and objects in command sentences, which widened the player's options. Infocom continued to improve its parser through the *Zork* trilogy and the many other prose quests the Massachusetts-based company sold in the first half of the 1980s.

Robert Woodhead and Andrew Greenberg had a radically different notion. With *Dungeons & Dragons* firmly in their sights, the design-programming tandem used a version of Pascal to create *Wizardry: Proving Grounds of the Mad Overlord.* Robert Sirotek and Fred Norman started Sir-Tech Software in 1979 in upstate New York to distribute it. The bulky manuals precluded use of the usual ziplock bag packaging, so this closest approach to electronic D&D met the world in an attractive carton.

Wizardry imports many of the rules Dave Arneson and Gary Gygax wrote for D&D, their fantasy RPG, like character classes, hit points, and experience. The player generates characters, assembles them into six-member parties, and dispatches these bands to the underground dungeon. In the creature-infested rooms and corridors beneath the castle, these bands fight evil, find treasure, and try to retrieve the artifacts spirited away by the tyrant of the title.

Line drawings, somewhat akin to vector graphics in appearance, show the world in first-person view. The player sees through the characters' eyes and issues movement, artifact manipulation, and combat orders with the keyboard in this turn-based contest.

The Digital Auteurs

By the early 1980s, three hardware brands ruled home computing: Apple II, Atari 800, and Commodore VIC-20. As usual, hardware molded the software business that supported it.

Under the aegis of Steve Wozniak and Steve Jobs, Apple burst on the scene with the slogan "Computers for everybody." Its tape-drive machine instantly captivated hackers—and ran afoul of the government's RF (radio frequency) emissions regulations. After a little retooling and the addition of Wozniak's microcomputer disk drive, the Apple II offered color computing and a 48K memory for less than $2,500. (The price fell steadily to about $1,200 at the time Apple introduced the IIc.)

Open architecture helped make the Apple II the first choice of computer hobbyists. Many of today's great designer-programmers learned their craft on an Apple II. Interplay's Bill Heineman was so fond of his that he carried it in a sack everywhere he went and often used it as a pillow! The Apple was the dream machine—allowing game design aspirants to work in the privacy of their bedrooms on their personal bids for gaming stardom.

Ironically, the Apple's limitations made it easier for solitary dreamers. Though Apple games had much more sound and color than the drab TRS-80 programs they superseded, the hardware didn't permit intricate artwork or lush music. This proved ideal for individuals willing to invest long hours learning programming. If state-of-the-art music consisted of little more than a string of beeps and boops, most elected to write it themselves rather than find a musician or composer.

Market entry was never cheaper than in 1979-82. There weren't dozens of magazines, so companies that could have afforded blanket advertising had no way to wield that advantage against one-man shows that couldn't have afforded an extra postage stamp. Mail order buyers weren't picky about packaging either, so entrepreneurs didn't have to worry about expensive box cover art and carton printing. Distribution was primi-

tive; game-lovers frequently bought through the mail, direct from the gamemaker.

The Atari Program Exchange (APE) gave amateur designer-programmers a focal point for their activities on the Atari 400/800 family of systems. Through company-financed catalogs, many games on disk for the system reached Atari's whole hardware base. A royalty system enabled Atari auteurs like Greg Christiansen (*Caverns of Mars*) and Chris Crawford (*Eastern Front*) to earn nearly $100,000 on ground-level hits.

Commodore's VIC-20 was small, cheap, and not very powerful, so naturally people loved it. Users could compose and run small programs (the only kind that fit the tiny VIC-20), which encouraged a lot of amateur game creation. Commodore never learned the knack of supporting its equipment with first- or third-party software, so part-time game inventors filled a gap.

The overall climate in computer software during the early 1980s favored the auteur principle—one person's vision guiding every phase of the game from concept to documentation. This view compared a game to a novel and decried game development by committee. Though games aren't made this way today, little more than a decade ago do-it-all designer-programmers made remarkable breakthroughs in every aspect of electronic gaming. Besides the ones already mentioned, this era produced groundbreaking designers like Bruce Artwick (*Flight Simulator*), Bill Budge (*Raster Blaster*), Bruce Carver (*Beachhead*), Barry Morstain (*MicroLeague Baseball*), and Richard "Lord British" Garriott (*Ultima*).

And you could get rich. Auteurs lucky enough to sign a favorable contract could earn as much as a million dollars. The key was the royalty percentage. Publishers that based deals on sales of ten thousand units found themselves signing away six-figure checks when the title sold over a hundred thousand. Designer-programmers like John Harris (*Jawbreaker*, a *Pac-Man* clone) who scored a best-seller could finance their own software companies. Alas, this kind of success was often as transient as winning the lottery. Auteurs without the business sense of a Ken Williams or Doug Carlston spent their money unwisely, and some wanted nothing more than to enjoy their windfall while it lasted.

Joining the Game-Team

Instead of relying on omni-talented individuals, several companies put together design/development teams, but none did it better than Electronic Arts, which was founded in the 1980s by Trip Hawkins, who'd puttered with computers during the 1970s. EA popularized this new way of producing computer entertainment software, and soon the vision of the designer-programmer as an electronic novelist was replaced with a team model derived from motion pictures: though one person may exert a tremendous creative influence, each movie is the result of many experts and specialists pooling their energies.

Hawkins, who cut his professional teeth at Apple, wanted to expand hardware capabilities. He likened a game to a painter's blank canvas. The more memory, graphics, and sound power in the hardware, the larger the canvas for software-makers. The sheer size of the canvas would force publishers to give up their traditional methods of design and development. EA would stand the auteur concept on its ear. Hawkins felt even the best game creator couldn't do every job well once the content of each game element reached a certain depth and complexity. As previously mentioned, anyone could write instructions for a charge riff or a warning buzzer, but composing a full musical score demands a higher order of talent.

Electronic Arts analyzed a software auteur, figured out what he or she did best, and then surrounded the person with specialists who could upgrade the weakest aspects of the design and development. The EA system improved entertainment software for two main reasons. It freed the finished product from the personal limitations of its principal inventor. An auteur who's a great designer but an indifferent programmer tends, perhaps unconsciously, to keep design elements simple. Conversely, the best programmer in the world may not necessarily conceive of the most interesting game, and a lot of titles are little more than eye-popping, semi-interactive demos for the cyber-cognoscenti.

The team structure also got the most out of the best people, with no one wasting their time and talents on things they did not do well. At EA, the main creative person could work on several games at once while concentrating only on his or her specialty. The team concept allowed the auteur to infuse his or her magic

into several products, instead of compelling the person to grind through the entirety of a project without help.

Not everyone adjusted. For many years Chris Crawford clung stubbornly to his lonely pursuit of software perfection before reluctantly bringing in graphic artists and others to buttress his fine design and coding work. Auteurs who couldn't learn to work within a team tended to have trouble finding work on top-drawer projects after the mid-1980s.

Electronic Arts also changed the industry in another, extremely significant way. Instead of maintaining a large in-house staff, EA producers hired an outside development group or signed an ad hoc team made up of both in-house personnel and independent contractors for each game project. When the game was finished, the team was broken up, and independent designers would either move on to other projects with other companies or be rehired for a new game. This helped to popularize the independent development house.

Before EA, companies kept designer-programmers on staff and tried to do everything in-house, which was cheaper and made it easier for management to ride herd on the creative people, which seemed the only way to get this unruly bunch to finish the work on time.

However, in practice, companies found it hard to control their in-house designers, whom they saw as a "necessary evil." Like the doctors in *M*A*S*H,* their usefulness made them hard to discipline. You couldn't suspend them because then the work wouldn't get done. You couldn't fine them for their transgressions because they didn't need much money to sustain their jeans-and-burgers lifestyle. Besides, they were such sensitive souls that rough treatment might freeze them at their keyboards.

And then the work wouldn't get done.

The product design/development people didn't keep regular hours. They didn't wear suits. They ate smelly food in their cubicles, pinned weird posters and newsclippings on the walls, and sometimes got sidetracked by "interesting" problems for weeks at a time. Despite this, designer-programmers logged marathon sessions to bring games to market by deadline. Marketing and financial people might have griped about their unruly, ill-bred colleagues, but they also knew these crazies were indispensable.

Corporate image aside, in-house development had a serious flaw. As the amount of code in an entertainment computer program rose, each game category began to become more of a specialty. It wasn't unusual for an Atari 2600 designer-programmer to do a mixture of sports, arcade, and adventure games. Computer funware was much more complex than that. It took real sports knowledge to design a simulation that would pass muster with end users. A publisher that depended on in-house personnel either had to specialize in one or two types of games or they had to pay the salaries of a range of specialists, who may not always be needed.

The pendulum has swung between independent and in-house development several times since the early 1980s. It's a cycle. The big publishers cut in-house staff and hire independent developers. The developers earn lots of money and expand. The most successful become publishers in their own right or merge with an existing company. Then the scarcity of independent houses forces companies to find new designers, programmers, and specialists and bring them in-house to ensure the talent supply.

Similarly, most operations start by concentrating on one type of game. They have the personnel to implement sports or adventures or platform games, and they do as many of them as the market can absorb. But game categories are like clothing fashions and change every couple of seasons. Sooner or later, the developer must either branch out or fire people. The developer does not want to lose anyone, especially someone needed on a later project, so the company accepts commissions for other types of gameware.

The company gets a little larger with each new project. The monthly overhead increases right along with it. Management sucks up even more contracts to stay solvent. Since the number of new releases can't just keep growing indefinitely, the developer hits the wall and hemorrhages red ink. To survive they must gut the in-house staff and rely on outside contractors, the talent scatters to the four winds, and new small development houses sprout.

Mining Our Business

Bill Hogue, one of the computer auteurs, developed a game for the Atari 400/800 called *Miner 2049er* in early 1983. He distrib-

uted the cartridge under the Big Five Software banner, but he didn't have the know-how to do other versions. When the Atari 400/800 went into steep decline two years later, that would have been the end of *Miner 2049er,* just another one-system wonder. "Great game," a few with long memories might say when nostalgia pushed its title into a conversation.

Enter Barry Friedman. Active in the industry since the late 1970s, doing everything from cartridge manufacturing to box art production, the master deal-maker conceived a bold plan around *Miner 2049er.* Instead of spending a year learning to program on the Apple II as well as he could on the Atari 800, Friedman licensed editions of *Miner* for other platforms. But Barry Friedman seldom stopped at half measures. Before he was done with in-person visits and his patented phone conferences, Friedman had a worldwide network ready to produce one or more versions, or SKUs (stock-keeping units), of *Miner 2049er.*

Publishing the same product for all popular systems drastically improves the bottom line by amortizing packaging, advertising, and promotional costs over several retail SKUs.

Multisystem publishing meant a new world for game designers, but the best way to accomplish this wasn't immediately clear. Some companies opted to develop each version independently of the others. This method was preferred by some British publishers such as TyneSoft, which helped put companies like TyneSoft out of business. Most US outfits quickly realized that independent production wasted money and often yielded games too dissimilar to justify a common title.

Another possibility was using a conversion house, an independent production group that specialized in "porting," or converting, games to other systems. These second-line operations couldn't design a commercial success, but they learned how to copy existing games very faithfully.

Unfortunately, a conversion house needed something from which to copy. Simultaneous publication across many platforms was out of the question and staggered product introductions increased the duration, and therefore the cost, of retail and consumer advertising. The method that worked best was for a company to organize several coordinated implementation teams, who would work separately and simultaneously from the same blueprint.

It didn't take many projects to discover that the standard two-page concept document wasn't enough to guide the separate implementation teams. Such cursory overviews left too much open to interpretation. Too many games ended up with key differences among editions. Programmers had always preferred for the specifics of design to be left to their own discretion, since this gave them latitude to insert their own ideas and modify others without regard to what management wanted. In those years, many project bosses had no techno-savvy. When a programmer shrugged and said, "Oh, it didn't look right, and it wouldn't have worked anyway, so I changed it," they couldn't tell if the analysis was correct or just an ego-spawned rationalization.

The best solution: Hire someone to write a complete design specifications document. This gave each implementation team exhaustive instructions. Some maverick coders still insisted on tinkering, of course, but putting a designer in charge of content minimized the discrepancies.

Working from a detailed blueprint also avoided another hazard: programmers usually favored one machine over the rest, and they designed software to optimize operation on their favorite platform. But platform-specific games were becoming a thing of the past, and a designer was needed who wouldn't favor particular hardware. And in this way, the job of game designer as it is known today was created.

The Second-Generation Video Game Consoles

Once Atari and NAP established the video game business, other manufacturers wanted to compete for the billion-dollar pie. Most of these systems flopped. They lacked the necessary support of a strong library of well-designed cartridges. Nobody bought the hardware because no one wanted to play the software.

Toy makers who tried to enter the video game market found that their wham-bam marketing strategy—which focused on making a one-time sale—didn't work, and it killed pretenders like Emerson. Gamers are more like hobbyists, who spend a lot of money on their special interest over an extended period of time, and they expect to form continuing relationships with manufacturers. The question prospective game-console buyers ask most is, "How many new games will there be?" "None" or even "We'll

see" are not the answers gamers want to hear. A system that isn't growing is dying, and gamers seldom buy anything that reeks of imminent obsolescence.

Mattel's Intellivision was the first true rival to the 2600 and the O^2. Although most famous for its Barbie line, Mattel wasn't a newcomer to entertainment electronics. It had dabbled in portable electronic games as early as the mid-1970s. The portable units were design marvels that made race contests and football simulations out of a few well-placed and well-programmed LEDs.

Mattel learned a lot about electronic games from its portables. Sales always climbed when hand-held technology forged ahead, so Mattel applied the idea to consoles. Instead of a platform on the same level as the entrenched brands, Mattel backed a next-generation machine, even though it had to charge more.

Mattel discovered an audience that Atari and NAP had virtually overlooked—adults. Intellivision broke video games out of the "just for kids" ghetto. Moreover, this group of consumers was willing to pay a premium price to get what it wanted. Although Intellivision's 1981 list price of $299.95 was almost twice as expensive as a 2600, it generated $250 million in sales the first year.

Was Mattel's decision to go for an older audience planning or necessity? Very likely the strategy had elements of both. Much of Intellivision's success resulted from Mattel's canny analysis of its design strengths and weaknesses—and the company's ability to showcase the good points in a way that minimized the soft spots.

Mattel didn't chase the youth market because it couldn't hope to catch it. The hardware wouldn't run the kind of games young players wanted—lightning-quick shooters. Intellivision couldn't process fast enough for high-speed arcade contests, but Intellivision's graphics and sound put the Odyssey^2 and Atari 2600 cartridges to shame, and these things won the allegiance of adults. The more complex on-screen images satisfied older players who wanted an enjoyable play-experience more than a hand-eye challenge, who were attracted by Intellivision's eye candy but were perhaps a little less confident about their own reflexes. The superior sound and graphics also enabled Mattel to produce the first viable team sports games. In addition, the unit itself looked more like a stereo component than a toy.

Mattel's TV advertisements, starring George Plimpton, sold player-satisfaction, not frenetic excitement. The commercials unrolled at a leisurely pace, with the mild-mannered intellectual with the cultured accent, then at the peak of his *Paper Lion* fame, enumerating the Intellivision's points of superiority in side-by-side comparison with the field-leading 2600.

The unique Mattel controller opened design vistas, too. Instead of a joystick, the Intellivision's controllers combined a direction disk—similar to the later Sega Master System and Nintendo Entertainment System—with four action buttons and a monoplanar numeric touch pad. This was sophisticated control compared to the stick-and-button joysticks used with other game units.

Intellivision cartridges came with thin plastic overlays that slipped into place on top of the touch pad. Early titles assigned functions to the pad that were not easily handled by the direction disk or buttons. In *Baseball*, for instance, pressing the drawing of second base on the pad caused the fielder with the ball to throw it there.

The touch pad, at first hailed as an innovation, ultimately disappointed designers. It was a fine concept that was sabotaged by poor implementation. The monoplanar construction made it impossible for a player to pick out the "hot spot" buttons by feel while watching the screen. The constant shifting of attention from the screen to the touch pad threatened every user with whiplash. Raised buttons might have eliminated the control problem, but Mattel didn't want to raise the cost of an already expensive system by revising the hardware design.

Once producers realized that the touch pad was a surefire mistake, it wasn't featured during game-play and became little more than decoration. Yet they couldn't bear to just leave it unused. For some reason, Mattel felt obliged to pretend that the touch pad still mattered. Virtually all Intellivision titles made token use of the touch pad. Gamers keyed in the number of players and other similar options before the start of play, when nothing important was on the screen. Once the game began, the touch pad went silent.

Until the Intellivision era, the medium wasn't congenial to athletic simulations. *Boxing* and *Tennis* (both by Activision for the 2600) made some headway in reproducing individual sports—and, of course, video gaming started with a sport, *Pong*—but the

team sports cartridges were horrendous. The hardware couldn't reproduce even an echo of diamond, court, gridiron, or rink action.

Games such as Atari's *Baseball* for the 2600 didn't look or play like the real sport. They were symbolic, not literal, representations of reality. It was baseball because it had a diamond with four bases in the middle of the screen, but nothing else made it feel like baseball. A chorus line of one to three linked players moved as a group with the joystick, and they did everything from catching flies in the outfield to making put-outs at first base.

The Intellivision could present fairly detailed animated figures. Mattel's sports cartridges *looked* right. They had accurate playing fields, the correct number of players on each team, and strategic nuances drawn from the actual sport. They weren't just action games; they simulated both on-field action and coaching decisions.

Mattel also brought licensing to the forefront. Applying methods that worked well in its toy business, Mattel endeavored to connect each cartridge to some authenticating personality or organization. Instead of a generic hockey game, it licensed the National Hockey League. The name on the box separated the Intellivision cart from run-of-the-rink puck programs and bestowed an aura of authenticity. Until Mattel bought licenses for its sports titles, most licenses were associated with coin-op hits moving to the home market.

Coleco produced the Golden Age's last great hardware system, ColecoVision. The company, started as the Connecticut Leather Company, a manufacturer of cap pistol holsters, had moved into home video games at roughly the same time as Mattel with its Telstar home arcade. Telstar's triangular game packs fitted somewhere between the hardwired devices and true cartridge programmables.

As with Intellivision, ColecoVision was a next-generation system whose hardware capabilities determined the composition of the software. Like the coin-op video game systems of the early 1980s, it offered fairly simple backgrounds but exceptional character and foreground graphics and animation.

That's why Michael Katz, the top gamesman at Coleco, put the major software effort behind home versions of coin-op machines. They not only worked best on the ColecoVision but Katz felt that

action games, leavened with real-time strategy, would attract sophisticated gamers who had gotten itchy about the blocky graphics and primitive audio of other platforms. Coleco's packages often showed the coin-op machine on which the home cartridge was based to underline the connection between its system and the amusement centers.

Making the arcade-oriented policy work wasn't easy. Atari and third-party 2600 publishers had already signed up most of the popular pay-per-play titles. A relative latecomer like Coleco didn't have a lot of "no brainer" licenses to consider. That didn't stop Katz and his sharp-eyed spotters. They combed the small companies and came away with lesser known but excellent properties like *Venture* and *Lady Bug*.

Coleco's strategy was successful, and more than any other cartridge game before it, ColecoVision enhanced the status of coin-op game designers by validating the financial worth of coin-op titles. Henceforth, amusement machine manufacturers jealously guarded the sanctity of their rights. What the coin-op manufacturers protected, they respected. That increased the prestige of the designers who generated the content of those suddenly valuable properties.

ColecoVision's licensing blitz also put an end to stealing ideas for home games from the arcades. Designers who could invent new themes and play-mechanics without that crutch became more valuable.

One of Coleco's best-selling originals, *Smurf Rescue*, contains several notable firsts. The most important is that this side-perspective, multiphase action contest is in many ways the ancestor of the Mario cartridges that propelled Nintendo to worldwide fame. The machine's limited memory meant that *Smurf Rescue* couldn't offer levels as numerous or extensive as contemporary side scrollers, but it featured remarkable, cute graphics and slightly more varied play-action.

Smurf Rescue doesn't have the now-customary bosses—particularly difficult foes that typically provide the last and greatest challenge at each level of a game. Instead, the smurf hero jumps into the next segment without pause or interlude. When he comes to the end of his arduous journey, the grateful smurf princess kisses her rescuer.

Which leads to *Smurf Rescue*'s other claim to design fame: it is the first game with a modern Easter egg, which is an image or sequence embedded in a game that does not activate during ordinary play. To activate the image, the user must perform one or more actions that have no other purpose in the play of the game. (The name derives from the tradition of the Easter egg hunt.)

There were Easter eggs prior to *Smurf Rescue*, but they consisted of the designer-programmer's secretly encoded initials. The practice started as a protest against company-enforced anonymity for game creators. Burying a name or initials gave designers a way to identify their work in the absence of proper crediting.

The Easter egg in *Smurf Rescue* is just as subversive, but a lot more clever: If the hero hesitates before the final series of jumps to the princess' perch on the giant skull, the princess offers further encouragement. To make sure the hero completes his mission, she throws off her clothes!

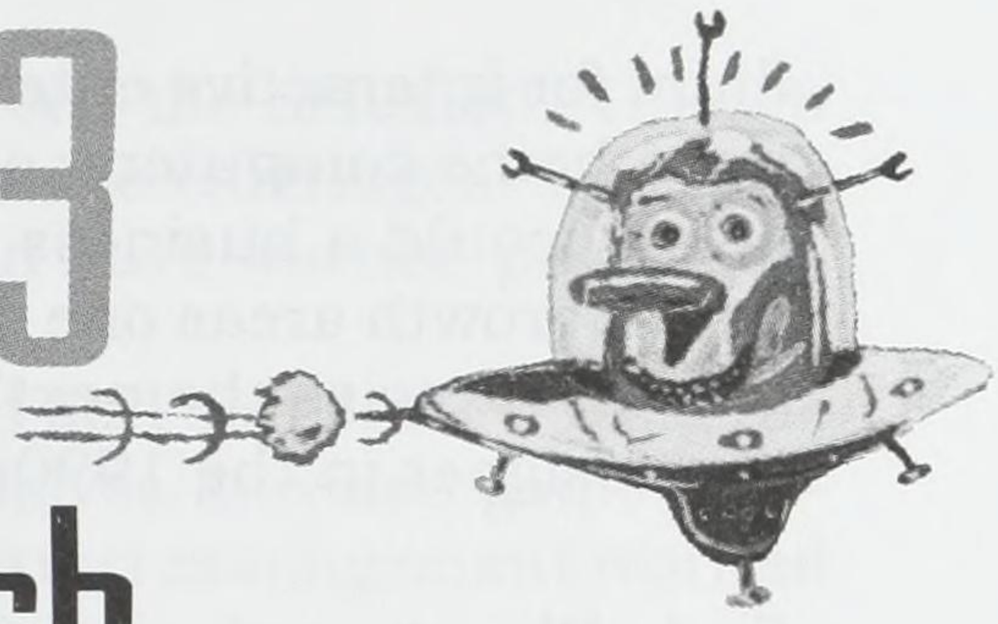
A less savory Easter egg appeared in an illustrated computer adventure of the same period. If the user kept typing after the character died, the text shifted to the circles of Hell where the character engaged in sex with a dog.

The Show Never Closes

At the peak of the Golden Age in 1983, it seemed like the good times would roll right into the twenty-first century. Consumers displayed an insatiable appetite for hot new games, and the team concept had fixed the short supply of talented gamemakers, at least temporarily. Some thought game consoles would become as common as VCRs.

The optimistic predictions of game company executives at the summer 1983 Consumer Electronic Show (CES) now look ridiculous, but even the most grandiose forecasts didn't seem that outlandish in the context of five straight years of double-digit growth. However, as we'll see in the next chapter, all that had come together was soon to fall apart.

Chapter 3



The Great Crash

The complete and utter collapse of the US video gaming market in 1984 is the most significant event ever to befall interactive electronic entertainment. Nothing, not even the introduction of the Atari 2600, has had a greater impact on the business and the game-lovers it serves. Almost overnight, to the complete surprise of nearly everyone, the boom in cartridge systems went bust, and the cartridge companies that didn't go bankrupt or get out of gaming entirely were severely humbled.

More than ten years after the cataclysm, it still dominates the thinking of every management executive, retail buyer, distributor, and creative person in the field. When Sega and Nintendo led the late 1980s revival, residual cynicism about anything to do with cartridge consoles posed the most formidable obstacle.

For a couple of years after the Great Crash, the word "game" was anathema to many publishers. Memories of the Great Crash were so painful that they tried to clothe pure recreational titles in the reassuring respectability of "edutainment." Some products had educational elements, some didn't, but all were forced into the edutainment pigeonhole just the same. One overzealous company advertised an arcade-style contest aimed at kids as educational because "it teaches the art of camouflage."

The Great Video Game Crash had an equally profound effect on game design. Because the computer side of interactive electronic entertainment experienced no similar disaster, in less than six months, the disk replaced the cartridge as the primary me-

dium for interactive entertainment. Electronic gaming veered toward home computers, and game design changed accordingly.

How could a business rank as one of the economy's most dynamic growth areas one day and become an inexhaustible source of bankruptcies the next? The answers helped define the role of the designer in the 1990s.

Early Warning

Designers and programmers were among the first to perceive the coming downfall. The mainstream media missed the signs, Wall Street misinterpreted the situation, and the major companies put up a big public relations smoke screen, but the cyber-artists got the message.

They couldn't ignore the staff trimming and perk rollbacks when sales fell from their 1982 peak. Designers and programmers were the first groups to benefit from the vast increase in the number of cartridge makers in 1982 and 1983, but royalty statements don't lie. When their royalty checks started shrinking, they *knew* something was seriously wrong.

In less than two years, from mid-1983 to mid-1985, video game sales plummeted from $3 billion to $100 million. To put that horrendous drop in perspective, Atari alone recorded $1.7 billion in sales in 1982! That's close to twenty times the dollar volume for the whole industry two years later.

Lesson One: Quality Matters

The industry's woes started with bad games. Very, very bad games. Publishers lost track of the importance of design in a mad rush to gobble up retail display space, and they shriveled consumer demand with a parade of the sorriest software ever plugged into a console.

Talk about an attitude! Nouveau riche companies thought they could sell anything. During the boom years (1979–82), they were close to right. Video games were the rage, kids loved them, and—so went the prevailing industry opinion—they would go on loving them forever. This comforting homily deluded manufacturers into a false sense of security.

Cartridge publishers tested the "sell anything" premise many more times than once too often. When a game everyone knew

was a clunker sold anyway, it reinforced the delusion. This, in turn, encouraged even more questionable cartridges. Consumers eventually rebelled, as logic should have warned publishers they must.

The majors, especially Atari, grew fat and corporate during the boom. They started to think of themselves as video game marketers, not entertainment producers. Atari management worried less about the cartridge and more about the box.

This view led Atari to undervalue development and emphasize advertising and promotion. The *Quest* series epitomized this philosophy. The four semilinked adventures were the subject of a contest in which the winner would receive a jewel-encrusted golden sword along the lines of Excalibur. The games, alas, were so undistinguished that the series petered out after the second title. Atari produced the last two mostly to finish the promotion.

The brain drain further depressed the quality of Atari cartridges. Designer-programmers deserted the Sunnyvale, CA, colossus to join aggressive start-ups or found their own development studios. Finally, after most of the gilt-edged talent had left, Atari got nervous and panicked. Company executives woke up to find that the creative cupboard was practically bare.

With unfinished games already presold into retail outlets, Atari had to ensure that there would be sufficient staff to get them out the door. To keep from losing still more key creative people, Atari instituted royalty payments in 1983 for designers and programmers. Until then, in-house personnel were salaried employees who didn't directly benefit from the games they invented.

The day Atari's first batch of six-figure checks hit the development department is one of the greatest days in game design history. Lack of proper game crediting remained a problem for years to come, but henceforth, video game designers had the same kind of participation in the success of their creations as computer entertainment software authors.

But Atari's move was too little too late. Consumers couldn't afford to be blasé about the deterioration in cartridge game quality. At $50 or more a crack, a bad game was an expensive mistake. Confidence in the video game publishers, once strong, melted away as once-trusted companies stung their most loyal supporters.

Atari's *Pac-Man* for the 2600 was a punch in the gut. It dealt a mortal blow to the publisher's image. *Pac-Man* was awful, and

months of high-pressure hype over its impending release made it seem even worse. The play-mechanic was faultless, of course, but the flickery graphics induced headaches. The quintessentially cute game came into homes in an ugly disguise and the audience repudiated it.

Bad as it was, *Pac-Man* was a software misdemeanor next to the 2600 versions of *ET* and *Raiders of the Lost Ark*. They were capital offenses. The cartridges shared the same basic program. Only the graphics differed. Gamers recognized the redundancy, though sometimes only after buying both of these heavily advertised carts.

All three games sold well initially, mostly because of hot licenses. But they shattered gamers' illusions that major brand names guaranteed quality. Despite large initial sales, none of these three carts had staying power. The surplus cartridges became landfill outside Albuquerque, New Mexico.

As video game design quality tumbled, consumers became more price-conscious. Not even the biggest companies could sell a cartridge for $50. The average price of a video game cart fell 50 percent in 1983 as fringe companies fielded discount lines. As sales worsened and small publishers went under, they took drastic steps to liquidate inventory on their way to bankruptcy court, dumping unsold merchandise. By 1984, bargain bins with $5 games were common.

Even design successes no longer satisfied consumers. Designer-programmers had maxed out the capabilities of the four leading consoles around 1983 and could no longer promise significant title-to-title improvement. Yet consumers continued to expect each new entry to eclipse the previous one. They didn't understand that it couldn't go on that way forever. Gamers gave the 1984 holiday season releases a lukewarm reception, which meant that few publishers had a megahit or two to balance all the misfires.

Video game development ceased when the market for full-price cartridges disintegrated. Why spend time and money on new games when existing ones aren't selling? A lot of video game creators found themselves looking for new jobs. Some migrated to computer entertainment, while others left the game business entirely.

Fuel to the Fire

The Great Video Game Crash was a disaster, but it needn't have deteriorated into the apocalypse it was, virtually killing off video gaming in the United States from 1984 to 1987. The media helped turn what could have been a temporary, though sizable, downturn into the Great Video Game Crash. It wasn't a conspiracy, or even intentional in many instances, but the print and electronic media amplified the debate and created a climate of overreaction.

Television led the way, since broadcasters had always felt they had the most to fear from video games. TV preferred to see cartridge games as a fad rather than as a legitimate new form of entertainment, since only video games had the power to occupy 25 million TV sets that could have been tuned to "Laverne & Shirley" or the "CBS Evening News." Cable was already encroaching on TV's billion-dollar preserve, and broadcasters had no stomach for another, even more immediate competitor. Television reported the industry's troubles with ill-concealed glee. TV grossly oversimplified the situation, reducing coverage to thirty-second sound bites. Instead of analyzing the problem, most commentators grabbed onto the same old tired stereotypes about video gamers.

On the other hand, newspapers and magazines had never had any reticence about singing video gaming's praises. If anything, print media oversold the whole idea. Embarrassingly fulsome articles ran without letup during the first couple of years of the decade. Every game publisher was a financial juggernaut, every executive a computer-age Henry Ford. Writers turned a blind eye to the field's imperfections and problems in their eagerness to communicate the excitement of the phenomenon.

But ongoing newspaper and magazine stories run in cycles. After a certain number of highly positive articles, it becomes progressively more difficult for writers to sell similarly favorable pieces on the same subject. Editors don't want to run the same story over and over again, and they also strive for editorial balance. At some point, the only fresh angles on an overplayed positive story are bound to be negative. The only way a writer gets ink is to unearth previously undetected flaws.

Positive stories about video games topped out around Christmas 1983. As journalists stopped behaving like public relations

flacks and began looking at the industry more critically, some were shocked to discover that, financially, things really weren't as rosy as they'd led readers to believe. Some felt conned, and they set out to unearth the "true" state of the industry just as the industry suffered its first shocks.

Financial analysts regularly overestimated the profitability and prospects of video game companies in the year leading up to the collapse. They didn't understand software or related technology. Video gamemakers looked good on paper because of their large inventories. Based on five years of steadily rising sales, the analysts reasoned, manufactured and ready-to-ship carts were bankable assets. Game publishers predicted yearly incremental gains based on putting out more titles every year, and writers repeated the optimistic message.

But a warehouse full of cartridges isn't the same as one stuffed with automobile tires. A bad game is not worth its stated value if the company can't sell it. Just like a bad movie is worth almost nothing, no matter how expensive it was to make, if no one goes to see it. No depreciation schedule could possibly reflect the real worth of a game after it had lain unsold for six months. It took the Great Crash to teach analysts a lesson: A game company's aggregate sales track record is a dubious indicator of future total sales.

It had never occurred to newspaper and magazine writers that video games were artistic creations, subject to the whims of public taste. When that became clear, their harsh reappraisals of the quality of video games, augmented by headlines about layoffs and bankruptcies, took a lot of the glamour out of video gaming.

Teenagers didn't need to read the financial page to get the anti-video gaming message. The public debunking of video games hit them hard. Over and over, the media told them that the video game "craze" was finished. And nothing is more uncool than yesterday's fad.

If mindless enthusiasm inflated the cartridge bubble, cruelly negative press soon burst it completely.

The Super Systems Stumble

Another contributing factor was the industry's total inability to get consumers to buy a new generation of hardware. The final

group of systems introduced before the Great Crash never built an audience that would have made them viable target platforms for software manufacturers. Although most hardcore gamers upgraded their equipment from their original 2600 or O^2 to an Intellivision or ColecoVision, they refused to make another jump beyond the so-called senior programmables.

NAP's Odyssey^3, the least ambitious of the projects, fared no better than the grandiose ones. Its modest objective was to provide more memory so that Odyssey cartridges could have elaborate, multicolored backgrounds.

Odyssey^3 should have aimed higher. It utterly failed to meet consumer demand for meaty, more visually exciting games. Brighter backdrops are nice, but the O^3 didn't offer better play-action or wider strategic options.

Atari, which always did things in a big way, failed *twice* with new machines in the time it took everyone else to bomb once. The 5200 (twice the 2600, get it?) reached the retail shelves and died in one continuous motion.

The 5200's cartridges were somewhat better than the 2600, but Atari mistakenly thought it could rely on the appeal of its previously published coin-op hits. However, most of them were now available for other systems, and there wasn't a new game like *Star Raiders*—something new, exciting, and unique to the 5200—to whip up enthusiasm. Lack of compatibility with the 2600 and unorthodox joystick controllers were the final nails in the coffin.

Atari tried again in 1984 with the 7800 (three times the 2600, get it?) and slammed into the brick wall of retailer apathy. The 7800 was a superior piece of equipment, and the controllers were back to normal, but the software library was uninspired. It didn't help that many of the best games were also available for Atari's own 400/800 computer system, which offered consumers many additional features and capabilities.

Again with the 7800, Atari trotted out the same tired list of titles. Sure, the implementation was better, practically arcade quality, but gamers wanted a new platform to have equally fresh games. By this time it was becoming clear that Atari had no in-house designers and it was not straining itself to connect with outside developers. Faced with the need to support a new system in a falling market, all Atari could do was pump out

the best home versions yet of the same old *Centipede*, *Pac-Man*, and the rest.

Upper-echelon Coleco executives, who preferred the toy business anyway, never got comfortable with cartridge video games. Computers, they felt, would be more prestigious, so they researched and developed the Adam computer. They didn't do the R&D fast enough, so the system didn't arrive in time for Christmas 1983 as scheduled. Few consumers noticed, since when it did come out there was little software and almost no games to go with it. As a promotion, Coleco awarded scholarships to children of families who bought an Adam. (Commercials suggested that not buying your kid an Adam was tantamount to child abuse.)

The Adam was huge, a bizarre expression of the Territorial Imperative. The modular components of this supposed wonder machine, each the size of a Macintosh Quadra, overflowed the largest desktops. However, since fewer than a hundred thousand Adams were sold, the furniture crisis never got out of hand.

Sick with a corporate malaise that even four years of Cabbage Patch mania couldn't cure, Coleco filed for bankruptcy before any of those little Adam users reached college. The Adam itself died before any graduated from junior high, though user groups still rallied round this lost cause.

Mattel outdid its rivals when it came to stillborn projects. The manufacturer held private showings for an Intellivision II, announced a perpetually delayed keyboard for the game console, and introduced the Aquarius computer. Three strikes and you're out. Mattel decided to stick to toys, giving up the video game and computer business.

Burning the Scrub

Foresters sometimes burn the scrub to make way for new growth. The Great Crash swept away all the outmoded hardware and substandard games. Eventually, after several years of measured progress, the richness and depth of electronic games jumped ahead rapidly when computers took over as the main game machines. Computer entertainment had always stressed these attributes, and the Great Video Game Crash temporarily removed most of the high-intensity kinetic games from the mix. When the balance swung toward enhanced content on computers, it ulti-

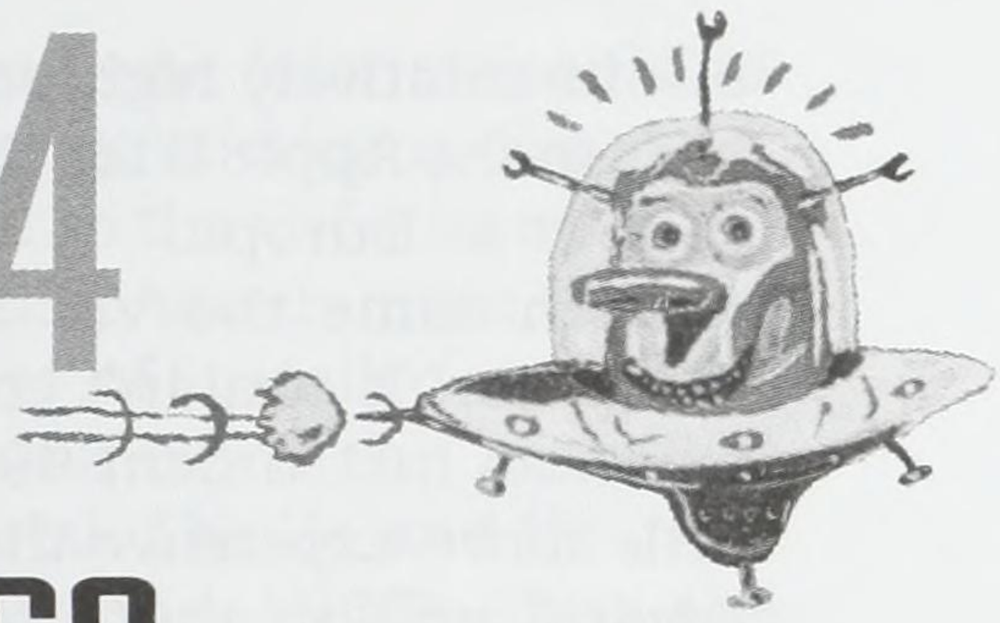
mately meant more and better work for designers, despite a few very troubled years.

Computer companies tended to treat designers better. Design/development teams had become well-established, so designers no longer needed to know programming. And computer software authors had always gotten more respect than those who had toiled at the hit-and-run cartridge outfits. Publishers put their names on the credit screen, in the documentation, and often on the package. Once Roberta Williams, Richard Garriott, Bill Budge, Bruce Artwick, and others became marketable names, due in part to the promotional blitz of Electronic Arts, publishers started to treat major game designers like stars.

Additionally, it is easier to hook on with a small development house than a big corporation like Atari. Everyone knows everyone in a small shop, and the organizational structure is often looser. An enterprising would-be designer might get a job in quality assurance or beta testing and, through daily contact, come to the notice of someone who could assign a design project.

The Great Crash left electronic gaming down but not out. Electronic gaming proved itself to be a multifaceted hobby, not as easily discarded as a run-of-the-mill teenage craze. Far sooner than anyone would have dared to predict, computer entertainment software rose out of the ashes of cartridge games. The resulting gaming industry was only about one-third as large financially, but if anything, it was even more creative than before.

Chapter 4



The Renaissance

So the press and the public buried video gaming. Few attended the funeral, but many gave smugly snide eulogies. Numerous articles and pundits categorized cartridge entertainment as a minor technological event like quadraphonic stereo. Many assumed that the death of cartridges marked the end of interactive electronic entertainment as a whole. The fad was over, trend-spotters chorused, bring on the Next New Thing.

They were wrong. The pundits confused the medium with the message, and though cartridges had failed, the desire for interactive electronic entertainment lived on. The nature of the games, however, did change. The computer play-environment required some design adjustments. Action-strategy contests and flight simulators supplanted frenetic shooters, and adventures reached a new high in popularity. In general, hand-eye coordination became less important than mental gymnastics.

The mid-l980s were the Dark Ages of video gaming in the United States, but design creativity flourished. This chapter examines the recovery from the Great Crash and how it caused the further maturing of the game design function.

Rising from the Ashes

Commodore Business Machines was the first company to create a computer that hit it big in the gaming world, the Commodore 64. An earlier personal computer called the PET did pretty well,

but its relatively high price and modest features made it second-best to the Apple II for most computerists, though the system did go over in Europe.

Then came the VIC-20, Commodore's first low-priced home computer. It couldn't compete with the Apple II, Atari 400/800, or Texas Instrument 99/4A home computers, but then it was a little more expensive than a cartridge console. Commodore sold several million machines despite its lack of visual muscle or memory capacity, and the VIC-20 soon went international (though it had to be renamed for Germany, where "vic" is evidently a rude slang expression). But its puny capabilities sentenced it to early extinction in the United States.

Then in 1982, Commodore mothballed the VIC-20 and brought out the Commodore 64. The slogan said it best: "Games and more." Built around a 6502 CPU, the same microprocessor at the heart of the Apple II and Atari 800, the Commodore 64 had better sound and graphics than machines costing three times as much. When the price went down to $599, it swept the computer world. Commodore's CEO Jack Tramiel kept cutting the price until it dropped so low that the C-64 cut into video game hardware sales. Its "games and more" marketing approach furnished prospective buyers with a fine rationalization for buying one instead of a cartridge console.

Game programmers hated it. Some, like Dani Bunten of Ozark Softscape (*Seven Cities of Gold*, *M.U.L.E.*), preferred the Atari 800, while hackers clung to the Apple II. "We had to program around the hardware," recalls Access Software's Bruce Carver. Lack of uniformity in the C-64's SID three-voice sound chip caused headaches. Commodore constantly tinkered with the guts of the systems, so developers couldn't guarantee that games would sound right on every component configuration. One night at a design/programming conference, boisterous developers bashed a unit with a sledgehammer to vent their hatred for the balky, yet money-making, C-64.

In 1983, Commodore hired John Matthias to direct internal software development. He'd produced three well-received action cartridges for the Atari 800 under the K-Byte name. The best was a well-animated permutation on *Berzerk!* called *K-razy Shoot-out.* But Commodore's internal working frustrated Matthias at every turn. He did publish a few memorable titles, but they mostly

arrived too late. His crowning achievement was *International Soccer*, a great action simulation of a sport North Americans don't love.

However, third-party operators flocked to the C-64 so enthusiastically that Commodore didn't have to do much more than go through the motions. As the C-64's base of installed machines rocketed into the millions, publishers banished the Atari 800 and the TI 99/4A. Even the Apple suffered. The IIe and IIc models kept the machine alive until the mid-1980s, but the Commodore's audiovisual edge proved insurmountable.

Ultimately, the Commodore's odd configuration proved its downfall. Programs could be fed into the C-64 in three ways: tape cassette, disk drive, and ROM cartridge slot. The cassette drive, similar to an audio tape cassette, was especially popular in Great Britain, where its economy made it the only cost-effective option. The Commodore's main flaw, slow access time, made loading a game from cassette agony for impatient Americans. The process was unreliable enough so that you couldn't count on a good load the first time. Getting a simple game to execute could take more than thirty minutes. A few outfits, like HESGames, made cartridges, but most put their trust in mylar instead of silicon.

The triple-level market hurt the C-64 because it artificially restrained designers. Too many US publishers demanded games that could be sold on cassette in Europe. This imposed limitations, especially on adventures and RPGs, which consequently never worked as well on the C-64 as on the Apple II.

The Commodore's meager RAM couldn't hold a full adventure. That meant multiple tape loads for all but the shortest programs. The secondary loading could interrupt a game for ten minutes, a definite mood killer. Additionally, the designer had to break down the plot into discreet episodes with no crisscrossing or backtracking; otherwise, the program couldn't be loaded via tape.

The multiload adventure is one of the most taxing design challenges in the history of electronic gaming. Segmentation produced games with the minimum possible number of in-game loads, but it took a very clever author to hide the linearity. Too often, it was foolish to even try. Only programs that loaded into RAM in one shot really worked well. This prompted the action game recovery that flowered with the return of the cartridge video game in 1986-87, a resurgence that was led by the Japanese.

For most of the eighties, the Commodore 64 became the world standard for low-priced computers. International software flourished, especially in Britain, and it's in the United Kingdom that that rest of the Commodore 64's story unfolds.

Software Anarchy in the UK

The British software market was volatile in the mid-1980s. Low unit prices and lack of video game competition made C-64 games rise and fall in popularity like singles on the pop chart. In the United States, the average C-64 game had a retail life of eight to twelve months. In Britain it was two months.

These factors shaped British software. The games were quick and dirty, slammed through development at warp speed to hold down costs. As a result, many games were ragged by our standards. Even top sellers in Europe, like *Outrun* (Sega), had to be redone to satisfy American consumers.

British companies dumped design responsibility on programmers. Many were otherwise unemployed teenagers who had a computing flair and would have worked for free. This glut of young people made it hard for mid-range professionals, who often had little empathy for the games themselves, to earn a living wage. Large companies paid royalties, but many small outfits bought games for flat fees. Needy teenagers willing to work for pennies frequently did just that, underbidding everyone else.

Sometimes the absurdly low fees boomeranged. A desperate teenager would lowball a project bid and start work, only to find it wasn't enough to live on. So the designer-programmer would go out and lowball another bid, get the advance, and start work on that one. Meanwhile, the first project went nowhere. A gamester with a knack for sales could hang up three or four companies before word got around.

US and British players had, and still have, different philosophical approaches to games. In the US, the ruling philosophy is "boot and bash," the belief that a game should be intuitively understandable and playable from the second the first images hit the screen. Americans love an intuitive interface, clear victory conditions, and a self-teaching first level. They also want unambiguous, logically arranged, and profusely illustrated instruction booklets in case they run into trouble. One ambiguous reference

in an RPG rulebook can ruin the play-session of an appreciable number of armchair adventurers.

Bob Jacob, founder of CinemaWare, once said that the perfect game design would let the user turn on the machine and play enjoyably without delay—and require no documentation. Jacob may have been speaking for effect, but it shows the commitment of US designers to smoothing the learning curve.

British gamers desired an almost completely different approach. Most British games omitted documentation to make players struggle to grasp the basics of a game, much less master the nuances. Anything that made the game take longer increased its perceived value. UK designers felt players identified more closely with games when they had to figure things out for themselves.

This was necessary, they felt, because the games themselves possessed few player-involving elements. British designers emphasized play-action. Their creations kept players busy and always guessing about what would happen next. While American designers labored over plots and histories—thinking this increased player identification—no plot was too flimsy for UK games if it served as an excuse for lots of jumping, running, and shooting.

Many UK programs confined graphics to a small active window with a decorative, static frame. Simple character images and repetitive background elements facilitated huge game-worlds. Sound was rudimentary, and no one had time to waste on frills like ornate introductions and cut scenes.

The side-perspective, action-strategy platform contest *Impossible Mission* (Epyx/home computers) wasn't a huge hit in the United States, though it sold pretty well. It bowled over British gamers and spawned a generation of imitations. UK designers reinterpreted its major elements in hundreds of action-adventures with hundreds, or even thousands, of simply rendered rooms.

These, then, were the factors that eventually hurt and doomed the thriving Commodore 64 near the end of the 1980s. As British publishers churned out games at a furious pace, too many lackluster products clogged US distribution channels. The glut of so-so disks drove down retail prices and forced manufacturers to accept lower profit margins on Commodore 64 software. Smaller profits didn't leave much money to retool the generic British games for the more finicky US consumer. Too many stateside companies just published the mediocre imports as-is.

Average games bore people. Lack of player involvement is what makes games average, so run-of-the-drive titles often share a short play life. The no-frills British game design style didn't hook Americans. Even discount prices couldn't make them palatable. American gamers were accustomed to buying six computer titles a year—not a game a week—and they didn't see enough value in the imports, even at lower-than-usual prices. After several strong years, the Commodore 64 went into decline as imports swamped the market. The manufacturer introduced the C-128, but no one introduced C-128 games for it.

By the late 1980s, the 6502-based home computers—the C-64, Apple IIc, and Atari 800/1200—were history. The Amiga, despite impressive specifications and popular applications in the graphics and music fields, did not emerge as the successor in the home market. The domestic Amiga software market started to decay in 1989, though foreign design/development groups gave the system outstanding titles like *Populous* (by Bullfrog for Electronic Arts). Active user groups persisted, but commercial development stopped. Almost all US computer game development after 1987 targeted the MS-DOS machines, with the rest specializing in the Macintosh. A few Amiga imports appeared each year through 1993, but the big projects and top talent all switched to MS-DOS.

The Designers Flourish

Computer games, which came to dominate the post-Crash gaming world, appealed to a different audience than cartridges: adults. And adults wanted intricate game-worlds and extended, involving campaigns. Adventures, RPGs, and strategy games filled the best-seller lists. Several sports simulations, including *Super Bowl Sunday* (Avalon Hill) and *MicroLeague Baseball* (MLSA), attracted large followings, too.

This new emphasis was a boon to American designers, since all of these genres require a lot of design work before anyone starts implementation. The content demands it. Publishers began hiring designers to create games, some of whom were former designer-programmers who couldn't code as well as they could invent the systems and mechanics.

This, in addition to hardware constraints, made "game engines" more practical, and game engines gave nonprogramming design-

ers their first real shot at creating games. A game engine is not a game but rather a program to generate them. Instead of modeling specific objects, the game engine designer models general cases that apply to many specific instances. A game program might have the full specs for a specific type of sword, but the game engine defines the category of edged weapons in such a way that it can handle all types of swords instead of just one. Most game engines are fairly rigid systems that require tightly formatted data entry. But once learned, anyone can follow the patterns, so publishers hired writers to produce sophisticated content.

The Japanese Invasion

No one told Nintendo Ltd. that video games were dead. Although the Japanese market has had frequent highs and lows, the market has never dried up completely. So even as the cartridge business was dying in the United States, the Kyoto-based manufacturer introduced the Family Computer System (Famicom) in Japan in 1984.

The Famicom, which was never introduced in that form in this country, is a Nintendo Entertainment System (NES) with some computer attachments. Basically, it plays video games. Japanese consumers, apparently unaware that the fad was dead, bought 6.5 million systems over the next eighteen months.

By late 1985, Nintendo was ready to try the United States, but most observers didn't think Americans were ready to try Nintendo. The market and gaming philosophy had changed dramatically. The Great Video Game Crash had engraved itself on the souls of its survivors. Manufacturers, distributors, retailers, designers, programmers, and even interactive entertainment critics were like victims of an awe-inspiring catastrophe, unable to see anything else. All had adjusted to the new world of computers. There was a lot less money in it, true, but the projects were more artistically satisfying. Both game-players and creators now eschewed the old cartridge platforms, programming, and action-style play.

Nintendo was faced with a less-than-congenial climate for a new video game machine, and they needed a gimmick to get passed retailer and consumer skepticism. What they decided to do was create a game-playing robot adversary that competed

against the human video gamer. The company tethered this device to the fledgling but otherwise normal Nintendo Entertainment System (NES), which it premiered at a 1985 press reception in New York City.

"See? It isn't a video game *at all!*" Bruce Lowry and his Nintendo of America cohorts told skeptical chain store buyers, in essence. "It is a robot toy! And robots are going to be hot! Hot! Hot!" And indeed, robots were hot in the mid-1980s, and retailers took a chance on the NES. Robots didn't remain hot very long, however, because soon everyone realized that no robot you could afford could do anything interesting, especially playing against one in a video game.

Designing games for the robot was a snap and the robot gimmick worked: American kids were intrigued enough to buy the new 8-bit NES, and they loved the regular video games it played. Once retailers discovered this, Nintendo quietly dropped the robot and the few games it had created to work with this peripheral. Sega of America (SOA), a competitor in Japan, followed Nintendo into the United States with its Sega Master System (SMS). The SMS was probably better hardware than the NES, but it managed to sell only one million units, versus 30 million NESs.

As always, software provided the decisive advantage, and it was at the bottom of Nintendo's success. The SMS may have had a more eye-pleasing color palate and a few other pluses, but Nintendo orchestrated a large array of sizzling cartridges. Taking a lesson from Golden Age Atari, Nintendo drew on its bank of Japanese titles and flooded the US market with as many games as possible. Nintendo kept the most pedestrian Japanese carts offshore, but there were still plenty of games to market here.

Nintendo and Sega pursued different marketing approaches; Nintendo came out ahead again. Sega published everything under its own brand name, in uniform packaging. Nintendo encouraged third-party licensees, but it also controlled the design and development process with an iron fist.

Unfortunately for Sega, this fostered the perception that more companies supported Nintendo. Sega looked monolithic and generic, while the large group of Nintendo third-party publishers suggested creative diversity.

Nintendo games targeted the audience better than Sega as well. The multilevel platform game, with a boss at the end of each

level, was perfect for eight to twelve year olds. Often slighted and underrated in their real lives, preteens felt empowered by the "conquer the game" mentality of the platform games. The process of jumping and running through the gauntlet of enemies and traps wasn't always exciting, but many young players got a thrill out of being the first kid in their class to blast through all the levels and discover all the Easter eggs.

Some adults criticized NES games for repetitiveness and lack of replayability. They missed the fact that Shigeru Miyamoto (the creative force behind the *Mario* line) wasn't designing cartridges for them but for the demographic group that actually bought the product. This is an important, if not always understood, principle of game design: A game succeeds only if it entertains the players. A game that's inaccessible is not a success, no matter how revolutionary the design.

A corollary is that a game only succeeds commercially if it finds an audience. The platform game concept, teamed with strong characters like Mario, reached its young audience in a way that no electronic game had equaled since *Pac-Man*.

Finally, Nintendo and Sega changed a key element of game design. Instead of joysticks, both systems incorporate controllers that assign movement to the left hand instead of the right. Aside from ambidextrous aces, most gamers can't produce the precise, intricate movement patterns with the left-handed direction disk that they can with the right-handed stick. Change the controller, change the game: 8-bit video games stress timing and targeting, not fancy on-screen movement. This is a typical example of how hardware puts *implied* limitations on software design. Ignoring such practicalities ruins many designs that read great on paper.

The "Graying" of Gaming

Eventually, Nintendo and Sega came out with 16-bit systems, Super Nintendo (SNES) and Sega Genesis, which are mostly souped-up versions of the 8-bit consoles. The canvas is grander, but it's mostly just more and better, not different. The Genesis and the SNES are logical progressions from the SMS and NES, just as the ColecoVision was of the Atari 2600. The 32- and 64-bit systems reaching the market now have not gone too far

beyond the 16-bit system yet, but they may well represent an even greater advance.

One effect of the 16-bit era that *does* influence design is the maturing of the audience. Tom Kalinske, president of SOA, reports that the average Genesis user is eighteen. That suggests two things: many 8-bit users "graduated" to the stronger systems, and 16-bit cartridges must speak to a sophisticated sensibility to succeed.

Video game publishers have found it difficult to cope with the wide age range of users. Nintendo ignored older players in favor of maintaining a Disney-like squeaky clean "family" image. Older players, therefore, bought the Genesis, which had sports and a wealth of nonplatform games.

Sega catered to this growing segment with grittier titles like Tom Zito's *Night Trap*, and this incited debate about violence in video games. In 1993, the US Senate held hearings on whether the federal government should regulate video game content. As too often happens, posturing critics with little knowledge of video gaming sounded the loudest alarms, and the debate turned into a farce. However, the hearings did show that many sincere people were worried, and the industry responded by developing its own video game rating systems. That's "systems," plural. At this writing, electronic gaming has reached no consensus, and several variations are plastered on package cartons. This is clearly an unacceptable situation, and a uniform standard will need to be developed in order to have a rating system that is something more than a marketing tool or a pointless infringement on the Bill of Rights.

The boom in multimedia, in high gear by 1994, intensified the age trend. The average Sega CD owner, says the company, is twenty-three. *Electronic Games*, a consumer monthly that covers all phases of interactive electronic entertainment, claims its average reader is twenty-four. Those who use CD-ROM for entertainment purposes on PCs and Macintoshes are even older, probably a median age of twenty-eight to thirty.

As Nintendo discovered at the start of the 16-bit boom, platform games don't captivate older gamers the way they do junior high schoolers. Nintendo has cut back on the level-on-level marathons, with the boss at the end of each level, in favor of a more diverse catalog. Sega, which never went as far with platform titles,

has done well pushing sports and shooters, though the Genesis wouldn't have been nearly as big a hit without *Sonic the Hedgehog*, a novel platform game, and its sequels.

Now multimedia is changing the rules all over again, making variety more than just the spice of life. It's the only route to survival and—dare to dream—prosperity.

The American Revival

Game design and development have revived in this country in the last four years by touting the "interactive experience." This has proven the best way to reach the US consumer, and our developers are the masters of this approach. Also, though five years ago most cartridges were designed in Japan, now more than half originate in North America. This has created a lot more opportunity for current—and future—game designers to ply their trade without emigrating across the Pacific.

Publishers have elaborated the original team concept to cope with progressively more powerful computer hardware. MS-DOS has breezed through four generations in about ten years. Today's 486DX and Pentium computers, fitted with powerful sound and graphics cards, do so much more than 286 machines in graphics, sound, and execution speed that the small design teams of the late 1980s would be overwhelmed by most 1995 projects.

Multimedia imposes still greater demands. It costs approximately $200,000 to take a conventional computer game from proposal to finished product. That's not chump change, but a multimedia title can easily ring up $2 million in expenses. That's what it takes to fill a disk with an interactive experience that satisfies finicky consumers.

Now producers break the design-development team into subgroups. Instead of one artist, a whole group of illustrators and animators work under the direction of an art director. Similarly, the design function is often divided among several people. There is still usually one person with design authority, but specialists lend a hand with on-screen text, modem interfaces, and other specialized areas.

Game Designing in 1995

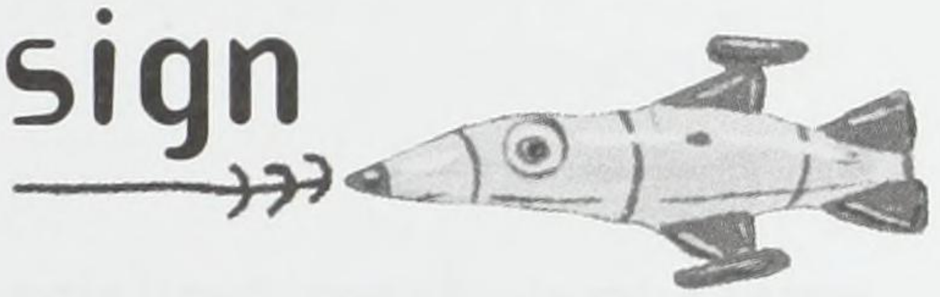
Multimedia and high-end computer software opens the gates for a new type of design specialist: authors from other media. It often makes sense to hire a screenwriter for an interactive adventure game. Most publishers still want a designer to create the game's structure and systems, but a dialogue specialist comes in handy for those lengthy cinematic sequences.

Does this diminish the designer's importance? Larger design-development teams lessen each participant's input, but this is a natural development, part of the maturation of interactive entertainment. The lead designer still handles all the major tasks, but now there are others to do specific jobs.

If anything, larger projects have increased the allure of good design. Games are now so complex that only a professional game designer can pull together all that content—and make something that's as interesting to play as it is to watch.

PART TWO

The Theory of Electronic Game Design

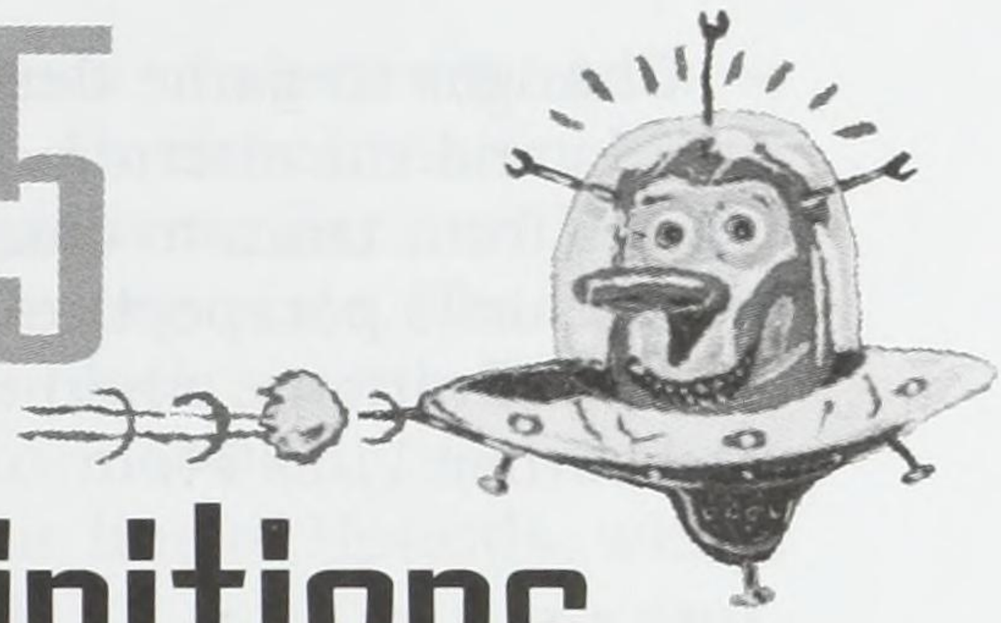


Don't skip this section! These pages are must reading for anyone who's serious about a game design career. It also has core insights into game design for those who just want to understand the behind-the-screens process of game creation. This is where we stop being interested bystanders and start exploring the inner workings of electronic game design.

The first step to becoming a professional designer is to understand the theories that underlie the practice. Without that keystone, any attempt to design an electronic game degenerates into a chaotic hodgepodge. Anyone can look at a bunch of games, cull elements from each, and patch together an ostensibly new game. If that was the totality of electronic gaming, designers would be unnecessary. All you would need is a good marketing department to spot the new hot game and clone it ad infinitum.

It has taken publishers a long time to appreciate the difference between haphazard game synthesis and true game design. Some publishers never grasp the distinction, and most of those end up with a catalog of unsold "B" and "C" titles. However, these theories have been around since chess, checkers, and Parcheesi were new. The general philosophy of game design is today a road map for the construction of a commercially viable entertainment product.

Chapter 5



Important Definitions

Exacting disciplines require a specialized vocabulary. Game design has acquired a dictionary full of terms. Some are as familiar to gamers as "playfield" and "sprite." Others like "port" (transferring a program from one platform to another without change) and "SKU house" (a developer that does only ports) are used mostly within the industry.

A third category is words which mean something different in electronic game design than in other disciplines. Everyone knows about the melding of the movie and interactive entertainment worlds, but a little-mentioned aspect is the vocabulary collision. A game "producer" does roughly the same work as a film "director." Some call non-interactive sequences in games "out-takes," but movie-makers reserve that word to mean versions of a scene which aren't used in the final cut of the film.

Inside Electronic Game Design defines words in one of three places. Terms with the narrowest applicability are defined upon their first usage in the text. Common, easily defined words are in the Glossary. Words that need an extensive, lengthy explanation are in this chapter.

Professors of the English Language bring a variety of attitudes toward their subject. So do self-diploma'd Professors of the Game Designing Language. Mine is that words continuously change connotative and denotative meanings. In consequence, any dictionary definition is a snapshot, true only for that moment in time.

Changes in game design will necessitate the invention of new words and the alteration of old ones. Furthermore, nomenclature varies from team to team, company to company. One designer's "two-thirds perspective" is another's "angled side perspective."

All definitions, whether in this chapter or not, are current as of December 1, 1994.

What Is a Game?

Armored with some perspective on the history of electronic game design, let's look at the universe in which a game designer operates. The obvious and necessary first question: What is a game?

To paraphrase Merriam Webster, a game is "any form of play, amusement, activity, or sport involving competition under rules."

While this is technically accurate, electronic gamers need a more specialized definition, especially since the orthodox dictionary entry ignores solitaire games. Before computers and artificial intelligence (AI), most games required at least two participants. Apart from card solitaire, which is better classified as a puzzle, solo gaming was impossible.

AI can create solitaire games that are bona fide competitions, and this is a significant evolution for games. Before computers, it took at least two players to generate a competition or to even present challenging obstacles to victory. Imagine *Monopoly* without rival real estate magnates! What makes it a game is the competition for choice properties and the ensuing bartering for more.

Not that nonelectronic gamers haven't tried to play alone when no opponent steps forward. A mid-1970s survey by Simulations Publications, Inc., showed that war-gamers play games alone more often than against human opposition. Their method resembles a benign form of multiple personality disorder. Typically, the solitaire strategist first goes all-out for one side. Then when the turn passes to the other force, the simulationist tries to thwart the assault he or she has just initiated! Perhaps it bodes well for these folk that computer technology has ended the need to manifest schizophrenic behavior just to refight the Normandy invasion or Waterloo.

AI expands the possibilities of games by turning many traditional two-player contests into excellent electronic solitaire contests. For instance, *Fox and Hounds*, played by two on a

checkerboard, is the ancestor of hundreds of electronic games, including all the side-scrolling shooters. Unlike chess, in which both sides have identical forces and objectives, *Fox and Hounds* is a competition between two dissimilar antagonists. The Fox has one checker and can move on any of the black spaces. The other side has four Hound checkers that can move only forward on black. The Fox tries to break through the line of Hounds, which attempt to herd the Fox into a dead end.

Designers have concocted many solitaire games by assigning the computer to direct the Hounds with AI instruction programming. The dangers Sonic the Hedgehog faces as he bounds along are equivalent to the Hounds in *Fox and Hounds*.

So, with a nod of respect to Mr. Webster, here's a narrower definition more suited to our needs: An electronic game is a competition involving one or more machine or human opponents, conducted under rules using interactive electronic technology. An electronic game must have a goal, one or more obstacles to the attainment of that goal, and a means by which the goal can be achieved.

The Elements of Gaming

There are three main elements to the definition of computer games—the goal, the obstacles, and the means of achievement—and if any are absent, then no game exists.

First, something must become the bone of contention, the result that gets the player to play in the first place. Many video game designers believe that the player's urge to get to the final screen is sufficient motivation to overcome the machine-directed obstacles, and millions of young cartridge fanatics would agree. Most adults prefer a more emotionally involved or contextualized competition. That's one reason why sports simulations and RPGs are so popular with older players. If there is no goal, the player has no true motivation to play beyond simple curiosity and no point of connection to the images on the screen. Though people love to win, they also want to feel that they have accomplished something, even if it's only within the context of the game.

Once a suitable goal has been established, it is up to the second and third parts of the definition—the obstacles and the means to overcome them—to provide the core of what makes a game a game.

The excitement of playing is derived from the intensity of the conflict between the players and whatever stands in their way. Obstacles generate challenge. Without barriers, the player would win without any particular effort. Gaming could be defined as the pleasurable expenditure of effort—so zero challenge equals zero entertainment value.

The gap between an average game and an excellent one frequently comes down to how intelligently the obstacles compete against the player-directed character. The worst miscalculations in this area produce the kind of challengeless game that people blow through in forty boring minutes and never boot again.

Then again, if it is impossible to achieve the game's objectives, if the player does not possess sufficient means to overcome the obstacles, playing is pointless. Most electronic games are zero-sum propositions, meaning that if they are not won, they are lost. But there are some games that defy this rule and can never be ultimately vanquished. Every round of *Tetris* gets harder until the game outstrips the player's abilities. Players attempt to create complete horizontal rows out of falling geometric shapes, and sooner or later, the pieces flash down the screen faster than the competitor can maneuver them into full rows.

Tetris, designed by Alexey Pajnitov, is an example of a game with gradually escalating difficulty but no endpoint. *Tetris* ultimately reaches a level that no one can handle, and at that point it ceases to be a game. Pajnitov calibrated the difficulty factor so cunningly, however, that it takes a very long time for most players to exhaust their full play-potential. *Tetris* delivers many, many hours of entertainment before players can no longer compete with it.

Chess is a near-ideal example of our definition of a game, and of a zero-sum game in particular. It's a competitive contest between two players (or machine opponents) who both begin with identical goals and strengths. The goal is to "checkmate" the opposing king, that is, to put the king in a predicament where it can't move without being captured. Both players have an array of pieces that both block the opponent's maneuvers and provide the means to victory. Players can take their opponent's pieces through the course of play, and in this way the balance of power eventually shifts until one player does not have enough obstacles to stop the other player from checkmating his or her king.

Wolfenstein 3D (id Software) is a Fox-and-Hounds-like confrontation between the player-controlled hero and a stronghold full of machine-directed defenders, who stalk the player relentlessly and fire at him or her at every opportunity. The goal is to retrieve the secret war plans, kill the bad guys, and escape. Enemy soldiers, attack dogs, traps, and hidden rooms stand between the gun-toting hero and victory.

Flight Simulator 5 (designed by Bruce Artwick for Microsoft) is a solitaire flight simulator with no combat nor any enemies. The goal is to learn how to fly a plane using this complex simulation. The obstacle is the extreme difficulty of the task itself, coordinating all the plane's systems to keep it whizzing through the air in the desired direction. The player competes against him- or herself in a test of skill, not against an opponent. In simulation games, winning is not a decisive moment; it is the successful achievement of a high skill level over the course of the computer's scenarios.

Since achieving one's personal best is a more rarefied concept, it's probably no surprise that noncombat simulators appeal to an older audience, perhaps mid-thirties. Mature players are more apt to enjoy this form of challenge. But just because this kind of game doesn't have as much conflict as most boot-and-bash games doesn't make it any less of a game according to our definition.

Simulations and Simulators

Two frequently heard, and misused, electronic gaming terms are "simulation" and "simulator." Game designers regularly veer into these allied areas, so let's nail down the definitions. An electronic simulation is a realistic, interactive electronic representation of a real or imaginary event or activity.

This definition may make some software marketing executives scream "Heresy!" They could care less about the definition. They like the word "simulation" because it has a solid, official sound to it, as if you were getting something more important than a "game." They find this image handy when, for reasons of fashion and other social trends, "games" fall out of favor with the public. Entertainment software is a fashion-driven industry. Game themes, play-mechanics, and descriptive vocabulary pass in and out of style exactly like hemlines and mixed drinks.

When "games" temporarily loses its chic with consumers (and the chain-store buyers who love them), publishers retreat to words like "simulation" and "edutainment." The redesignation causes only minor revision of the product, more of a cosmetic alteration. During the Great Video Game Crash of 1984, one company advertised an adventure game as educational because it provided practice in mapmaking!

The main quality that distinguishes simulations from other types of games is their approach to the subject. Simulations treat the material in a realistic and detailed manner, while regular games use symbolic representations that are neither realistic nor detailed.

The same subject can inspire a simulation or a nonsimulation, depending on how it's used. War is the theme of both chess and *Operation Crusader* (Avalon Hill/MS-DOS). The former is an abstraction of mass conflict, while the latter models the North African campaign during WWII right down to the day's weather and the movement capabilities of the military units. Real knights don't move like a chess piece. No one expects them to; the horsehead figurine is just a symbol.

The dichotomy between realistic and symbolic games arose before the Digital Age. The first games ever invented were symbolic games, contests such as Parcheesi and Go. Realistic games, though newer, began with such landmark titles such as *APBA Baseball*, *Tactics II*, and *Dungeons & Dragons*. Both happily coexist today in both electronic and nonelectronic gaming.

Realism is a delicate concept, especially in regards to gaming. In design, it means that play-elements combine to give the user the feeling that they are participating in an accurate, if vicarious, reflection of reality. Sometimes this reality exists only as a game-world, but it's presented with the same rigorous internal consistency as a simulation of a real-life situation. *Outpost* (Sierra On-Line/MS-DOS, CD-ROM) is a simulation that concerns homesteading on other planets, an event still decades or even centuries in the future. Designer Brad Balfour treats the situation with the same logic and precision that Sid Meier brings to *Colonisation* (MicroProse/MS-DOS).

Simulation realism is a tightrope. It must convey the essence of the situation without either glossing over important aspects in the name of playability or burdening the player with irrelevant complexities in the name of accuracy. The most common simu-

lation design mistake is literalism. As a creative artist, the designer must filter the experience through his or her perceptions. Only those elements that make players feel like they're immersed in the experience belong in the game.

There will always be an audience for hyper-realistic simulations of activities like flying an airplane or driving an Indy racer. In general, though, players desire the distillate of an experience. They want the fun parts, with just enough grit to make the vicarious experience convincing. Few armchair pilots, even the most fanatic, want to vacuum the crumbs out of the passenger seats or reupholster the pilot's chair. Those things don't have much to do with soaring through the wild blue yonder, and they would be excess baggage even in *Flight Simulation* or *Falcon*.

Adventure game authors often fall prey to simulation overkill. Some try to create a convincing illusion by breaking down action into the smallest possible components. A game that turns sitting up in bed, dressing, and leaving the bedroom into a major test of mental agility, like *Hitchhiker's Guide to the Galaxy* (Infocom), is the antithesis of realism.

The advent of virtual reality and total immersion gaming will force designers to reconsider realism many times in the next few years. Publishers will try the literal approach, at least a few times, but design theory suggests that even the cyberverse will make simplifications and omissions when doing so improves the core of the experience.

Finally, the division between simulations and simulators is the grayest of gray areas. Ad copywriters use them practically interchangeably. When used by designers, however, a simulator is a particular kind of simulation in which the player directly assumes the role and function of one or more people in the game-world. In a simulation the player has a third-person perspective, and in a simulator the player assumes a first-person perspective.

Desert Strike (Electronic Arts/Genesis, SNES) is a simulation, but *Commanche Overkill* (NovaLogic/MS-DOS, CD-ROM) is a simulator. Both are intricate real-time simulations of helicopter-oriented combat, but only *Commanche Overkill* puts the player into the cockpit. Instead of seeing the helicopter from an external vantage point, as in *Desert Strike*, the player sees what the chopper's pilot sees and must perform actions analogous to a real pilot's duties.

The simulator category has another noticeable family trait, in that the means often eclipse the ends. In a flight simulator, the goal is to learn how to pilot the aircraft, and the missions measure the degree of mastery. In other words, the objectives are secondary to the ability to execute the aerial maneuvers necessary to achieve them. *Flight Simulator* is the purest expression of the simulator aesthetic because it doesn't include military or commercial consequences. The player attempts to acquire expert control over the plane, not operate an airline or defeat an enemy.

A Puzzling Matter

Is a puzzle the same thing as a game, and are some games no more than puzzles? As we will see, "puzzle" is an elusive term in regards to electronic gaming. The dictionary says it's "a toy or problem for testing skill or ingenuity." Unfortunately, that terse definition is practically a puzzle in itself. Designers need a more elaborate explication, and I would offer this: An electronic puzzle is an interactive electronic experience that tests the participant's problem-solving ability. A puzzle has a premise, a procedure, and a solution.

Puzzles and games incorporate many of the same features. They are both interactive competitions in which the player tries to secure victory despite barriers. Despite the overlap, there's a basic difference, too. A puzzle has a definite and singular solution. A puzzle may be quite complicated, but its solution requires that each piece be put together in the correct order in the same way every time. While many electronic games incorporate puzzles into their play, not every puzzle is complex enough to qualify as a game. After all, once you've solved a puzzle, the challenge is gone.

Many games, including classics like *Shanghai* (Accolade/all platforms) and *Tetris* (Spectrum Holobyte/all platforms), as well as newer ones such as *Lemmings* (Psygnosis) and *Gridders* (Tetragon), use a puzzle play-mechanic. Most side-scrolling video games are multistage puzzles, which partially accounts for their low replayability.

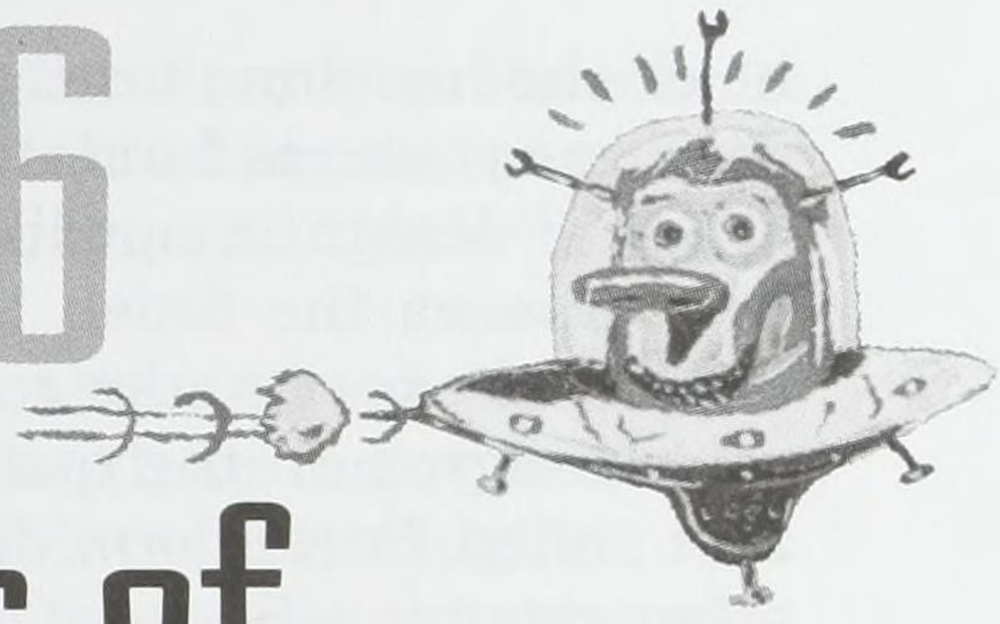
Designers disguise puzzles to minimize their inherent limitation. The structure of a puzzle forces linearity, which is the enemy of long play-life. If there is only one path through the game, few players will want to take it twice.

The simplest tactic is to assemble a very long series of puzzles. In such games, each solution is merely a prelude to the next challenge. This format is especially popular in video gaming. It's found in a myriad of platform contests, including *Claws Encounters of the Furred Kind* (Accolade/Genesis, SNES) and *Donkey Kong Country* (Nintendo/SNES). Although these games often let the player bounce back and forth on a level, hitting individual encounters in any desired order, each is a little physical puzzle. Puzzles are often embedded in the plot structure of adventure.

Another way to make a puzzle *seem* less linear *to the player* is to increase the difficulty over the course of play and keep the ultimate solution out of reach. Periodic victories reward the player, who gauges performance with a numeric score instead of aiming for completion. No human can beat *Tetris* at its top speed.

Puzzles also function as major plot triggers in many otherwise nonlinear adventures. In games like *Myst*, the player roams the world with every appearance of free will, but he or she must face each of those multilayered conundrums sooner or later to get to the end.

Chapter 6



The Aesthetics of Game Design

In the preceding chapter, quality is not mentioned in the definition of an electronic game. After all, it's not a requirement. If an activity satisfies the definition of a game, it's a game, whether it's fun to play or not.

However, quality, for those who play and those who design, is of utmost importance. We each have our own subjective standards of taste, but this chapter will attempt to describe how to distinguish the good from the bad, how to sniff out those elusive qualities that make the difference between an enjoyable adventure and one that keeps players up all night.

Game design itself resembles a game. The player (the designer) tries to achieve a goal (a commercially viable game) despite obstacles (hardware limitations, deadlines, and sometimes-wayward programmers). It's an unwinnable game if the player doesn't know the grading system. The prospective designer must study the standards of the judges before setting out to create a successful professional design.

This chapter explains the scoring system for electronic game designers. It's a guide to the victory conditions for the game design game.

A Subjective Universe

There is no universally accepted aesthetic standard for electronic games. There's no surefire way for a designer to know how well

he or she has done until the public casts its checkbook vote. The scoring system is fundamentally subjective.

Game design is not like the hundred-yard dash. The runner who crosses the finish line first doesn't necessarily win. Even supposed experts, the reviewers in the professional magazines, seldom agree on the qualitative merits of products. One magazine called *Earthworm Jim* "the game of the decade," while another rated it a B.

Game evaluation, like criticism of any other art, is not a science. At its best, criticism is an informed observer's subjective views explained in objective terms. The best critics strive to come to terms with their own biases and prejudices and allow for them. The worst critics praise and damn titles based on visceral emotional impressions.

Evaluating a game's technical proficiency is another matter. Quantitative norms exist for image definition, animation speed, command responsiveness, and so forth. The game animates at a given number of frames per second which is a specific number. Comparing the speed, in frames per second, of the average games helps us make logical statements on this matter. The artistic part of the game, however, can't be pinned down with comparable exactitude. But while two games that execute at the same number of frames per second are equally fast, the quality of the animated images remains open to debate. It's a matter of personal taste.

Art Versus Commercialism

Is there a difference between a good game and a successful one? At this point in the development of interactive electronic entertainment, the answer is a reluctant yes. A high-quality game is more likely to sell well, generally speaking, but exceptions occur. *Toejam & Earl* (Sega/Genesis) drew rave reviews from the adult gaming magazines and admiring comments throughout the industry. It sold moderately well, but it wasn't a Sonic-level hit, either. The cognoscenti loved it, but it caused barely a ripple in the mass market. Game design is an art within a profit-driven industry, and for now, even the best games are not considered successful if they don't sell widely.

Eventually, game companies will market directly to the whole country over the information highway. That will make it easier to target special interests with niche products, and this will create a huge change in the gaming industry.

The effect will be similar to the one cable and satellites have had on television in the 1990s. When the three major networks were the only national choices, they achieved their highest ratings (and attracted the most advertisers) by appealing to the widest possible audience. There wasn't room for esoteric programs that might pull only a few million viewers. Now, cable has provided so many channels and options that specialized services like ESPN, Nickelodeon, MTV, and the Sci-Fi Channel can thrive, even though they couldn't have cracked a network schedule a decade earlier.

Electronic gaming already has a few profitable niches, like ultra-detailed flight simulators, hexgrid war games, and statistical sports simulations, but these are still relatively minor segments within the total market. By 2005, military sims may have expanded into a mass market with niches for specific historical eras. Some enterprising publisher will launch a Civil War game club.

Until then, the issue remains cloudy. Game designers must still design with an eye to universality and accessibility. Those who hope to earn a living with their designs must aim at producing a game that will capture enough audience to earn a profit. A successful designer's work unites art and commerce, creating an excellent game that can pull strong sales.

Searching for Consensus

While individual standards of taste defy prediction, it is possible to describe the cultural standards of a particular place and time, even though these standards may change from country to country and from generation to generation.

Painting, which has been around for centuries rather than decades, is a good example of this. At one time, connoisseurs judged each painting on how closely it mimicked reality. A painting was considered excellent if it acted as a photograph does today, depicting people and objects exactly as they are in real life. Following this were periods when critics felt the best paint-

ings were semirepresentational (Impressionism and Surrealism) and even nonrepresentational (Abstract art). When Impressionism reigned, anyone painting painstakingly realistic portraits was considered out of step with the times, and their works were not highly regarded despite their technical virtuosity.

The proper question isn't "What is a good game?" It's "What is the currently prevailing conception of what makes a good game?"

There is no worldwide consensus. The fractionalized international software market offers many proofs. If there was planetary agreement, publishers would distribute games in substantially the same form in every land, changing them only to translate for different languages.

That doesn't happen. It may appear that every game invented in Japan comes to the United States, but fewer than half make the trans-Pacific leap in any form. Many Japanese games don't appeal to American tastes, and the ones that are imported must often be reworked extensively to harmonize them with our national sensibility. A nation's software embodies its popular culture and the expectations of its consumers.

This may seem like a debatable point if you subscribe to the stereotype of the alienated computer nerd. But don't confuse image with reality. The image is more applicable, if it has any validity at all, to the technical specialists, like programmers, who often learn their craft through years of devoted, intense, and isolated study. Their quest for digital knowledge may distance them from the social mainstream, but the connection persists. Even the most asocial, all-night-hacking programmer wasn't raised in the wilderness by wolves. They've watched television, listened to music, eaten fast food. Whether they embrace their society or rebel against it, mass culture is always there as a frame of reference and the reservoir of common experience.

Designers of games, and the people who play them, are likely to be more thoroughly socialized than the stereotypical computer nerd. After all, their expertise consists of the ability to transmute raw cultural inputs into artistic interactive experiences.

Designers spring from the society in which they live. They are educated by its schools, informed by its mass media, and nourished by its culture. What a designer brings to a game reflects their life experience up to that point. Shared experience creates

a society that holds essentially the same consensus on a wide range of subjects, even on the types of video games that are fun.

Thus, a matrix of basic attitudes underlies our own society's games. *SimCity* and *Doom* both embody aspects of the American psyche. *Doom* expresses an American pop culture stereotype, the vigilante gunslinger, which also includes John Wayne and Batman. *SimCity*, fittingly, derives from a more subtle American idea: Given a chance, we can show "those politicians" how to do it right.

An electronic game's concept, theme, graphic images, sound, and feel derive from the surrounding culture. Every choice made, from the first idea to the duplicated disk or cartridge, is influenced by the culture that binds gamemakers and game-players. If Japanese games have characters with big heads and enormous eyes, it's because that looks right to the Japanese. A US designer would never consider creating characters like that, simply because it wouldn't look right. Every country has mavericks and those whose special circumstances insulate them from the pop culture mainstream. No statements about a country's software style are 100 percent relevant to every design/development house within that nation. They are broadly applicable, however, to the full spectrum of a country's software catalog.

Despite disagreements among US designers about the details, there is widespread agreement about the big picture. The American designer's goal is to conceive of a game that offers the user a complete, enveloping interactive entertainment experience. This principle is so ingrained in the national psyche that few gamemakers would seriously question it.

In England, however, they will tell you that what counts is the interface and the play-mechanic. British games deal with situations symbolically, not realistically, so they don't expect players to submerge themselves in the game. Many British designers consider the lengthy introductions, cut scenes, elaborate rule books, and other accoutrements so important to US games as excess baggage.

The Interactive Experience

The prevailing philosophy of US game design derives from the literary criticism of Samuel Taylor Coleridge (the famous English

poet and trailblazing drug experimenter). The author of *The Rime of the Ancient Mariner* and *Kubla Khan* coined the phrase "willing suspension of disbelief" to describe the interaction between an enthralling novel or short story and the reader.

Coleridge propounded the idea that the writer's object is to transport the reader to the book-world so thoroughly and convincingly that the reader forgets that the world is just words on a page. Only by overcoming the words-on-a-page barrier can the reader fully experience what the artist is trying to convey.

Ironically, Coleridge is not an honored prophet in his own land, at least not within the British software community, which treats the enveloping experience concept as one of those unfathomable Yankee notions that don't apply to the Sceptered Isle. Willing suspension of disbelief is the cornerstone of the American electronic gaming philosophy. Americans prefer electronic games that simulate, rather than symbolize, reality.

The classic board games depict reality with symbols. A good example is the venerable *Monopoly*. The core theme of *Monopoly* is to purchase real estate, improve it with residential and commercial property, collect rents, and grow wealthy while crowding out less viable competitors.

That's the essence of the free-market economy. *Monopoly* is an interactive allegory of the American Dream. *Monopoly* says that anyone can buy a little land, build it up, and become as rich as the dapper old gentleman in the game's logo. It's an interactive Horatio Alger story.

Even the millions who bought the Parker Brothers hit during the Depression didn't imagine that *Monopoly* was an accurate simulation of the real estate business. The theme is an abstraction, and the play-mechanic has nothing to do with the way someone like Donald Trump buys and sells his properties. Atlantic City is not a big square with the municipal jail in one corner and the town's only parking lot in the other. No one hands you your annual salary just for walking once around the city limits. And what landlord doesn't dream of collecting rent from people who happen to be walking past a house or hotel? The play-mechanics of the game have no relation to the real-world activities associated with the game's theme.

Compare *Monopoly* to *Baron* (Blue Chip Software/PCs), a mid-1980s real estate simulation. The goals and the means to achieve

them are roughly the same, but it's hard to imagine two more different games. *Baron*'s design is complex and attempts to simulate real estate trading as close to the real thing as technology allows. *Monopoly* conveys the feel and the essence of buying and selling real estate, but its a simplified version.

The same progression is evident in the development of war games, from the nonelectronic *Battleship* to *Harpoon II* (360 Software/MS-DOS). The former uses military symbols in a game that could as easily be about rabbits hiding from foxes in the meadow. *Harpoon II* couldn't possibly be about anything except naval warfare.

Interest in realistic games has built steadily in the United States, but electronic gaming didn't turn decisively toward realism until 1984. After cartridges disappeared in the Great Crash of that year, publishers distanced themselves from the suddenly discredited video game. Computer-software companies feared that retailers would decide that all electronic games were in the dumper and drop disks as well as carts.

The industry's official line was that video games died because they had served their noble purpose. Game machines paved the way for home computers. The colorful, kinetic, and symbolic action video games were fine for their period, the rhetoric went, but now greater processing power would make possible deeper and more realistic simulations.

Artistic integrity isn't the only reason for advocating the philosophy of the enveloping interactive experience. There's a practical side, too. No games reach stores without one or more retailers authorizing an order. Since few retailers are gamers themselves, the subtleties of game-play sail over their heads. It's much easier to assault their senses with noninteractive atmospheric sequences, which require no more effort on their part than watching a *Star Trek* rerun.

How one creates that encompassing interactive experience is still a subject for debate. One widely held tenet says that the interface should be as transparent as possible. If the interface is intuitive and nonintrusive, designers know that players concentrate on the game-situation and forget the play-mechanics. If the interface is cumbersome, players get so involved in the operation of the game that goals become more distant.

This is illustrated by two games, the nonelectronic *APBA Baseball* and the electronic *MicroLeague Baseball*. Deeply similar, both

are statistical simulations of major league baseball. Although play-resolution is triggered by random numbers, neither includes luck or manual proficiency as a factor.

In *APBA Baseball*, human participants shoulder the data correlation, which the computer executes in *MicroLeague Baseball.* An APBA manager rolls dice to produce a random number, finds the appropriate result number on the batter's card, modifies it by the pitcher's rating, and then consults one of eight play-result boards.

The first thing that anyone does who plays *APBA Baseball* with any regularity is to memorize the result boards. It speeds up the pace, so there's plenty of positive peer reinforcement for those who can spout results without juggling the four, two-sided boards. The next thing that happens is that managers start analyzing the batters' cards. They unearth peculiarities that make two supposedly equal players in real baseball behave differently in the game.

For example, *APBA Baseball* decides which error numbers a batter has on his card according to his fielding position. This has nothing to do with real baseball; it's a way to allocate errors so that they occur with the expected variety and frequency. In *APBA Baseball,* it is only possible for the batter to reach second on a two-base error when a first baseman steps into the batter's box.

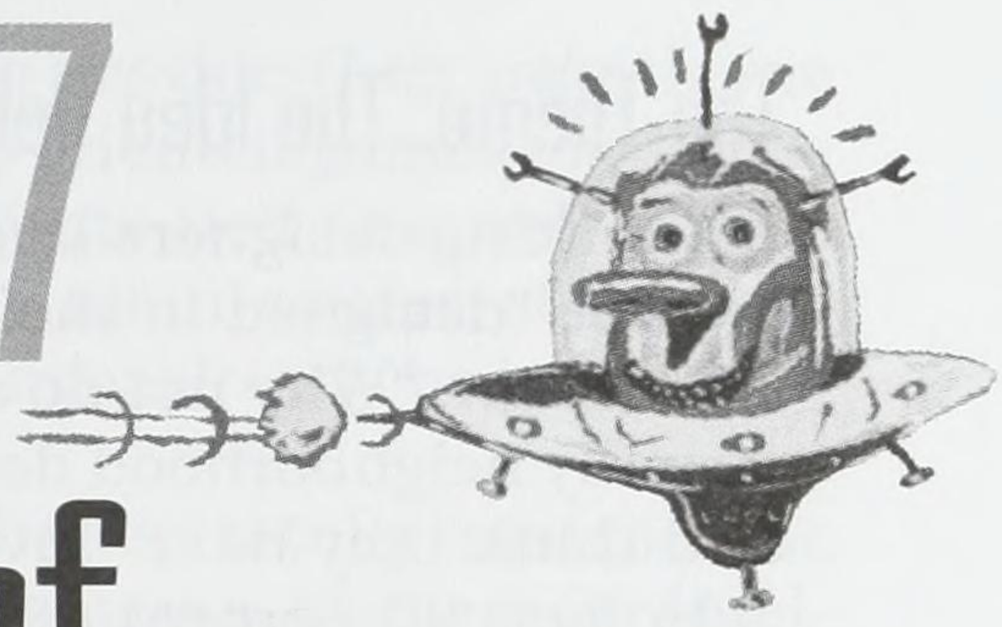
Canny managers consider these quirks when constructing teams, deciding on starting line-ups, and inserting pinch hitters. They learn to think less about the abilities of the real players and more about the play-result formulas.

MicroLeague Baseball buries the formulas deep in the program, inaccessible to players. Since managers can't see the numbers, they are forced to rely on their knowledge of the real players when concocting strategy. *APBA Baseball* is an excellent game, better than *MicroLeague Baseball* in some respects, but there's no question that the electronic game delivers the more true-to-baseball interactive experience.

Another example of the US drive to create more enveloping games is the death of the original parser-driven text adventures. They always fell flat on their interfaces. Typing in answers and playing guessing games to find a word the computer understands disrupts concentration and interferes with the willing suspension of disbelief. That's why the parser-based games have disap-

peared in favor of adventures and RPGs that use menu- and icon-based control schemes. In this way, the broad desire for a certain type of experience influences game design choices. Identifying what game-players want and how to give it to them is one of the major mysteries and challenges for game designers.

Chapter 7



The Anatomy of Electronic Gaming

In this chapter we will take the electronic game apart, much like an army recruit takes a rifle apart, over and over again, to become familiar with all its parts. A game designer must have an exhaustive knowledge of the individual elements of a game, otherwise it will be impossible to design one.

Players experience games as a totality, but designers build them one module at a time. A designer must write a complete, professional, specifications document, which separates the game into discreet sections. If the designer doesn't treat every component in full detail, the specs won't provide adequate guidance for implementation specialists. Then the producer won't be very happy. If that happens, he or she might decide to stop the milestone payments—checks handed out as specified stages are completed—which is a dark day for any freelance designer.

A game's components are produced separately but they must function together smoothly, and all the individual design decisions are interrelated. One of the things that makes a designer valuable is the ability to not only imagine the individual elements but to mentally fit them together. Like a composer who hears an entire song before the orchestra strikes the first note, the good designer plays the game mentally before the first pixel dances across the screen.

The Theme: The Idea Behind the Game

As working designers know too well, wannabes (people who have a game "designed in their head") often confuse the concept with the design. "Why not do a game about . . . ?" they burble at their friendly neighborhood designer. They utter their one-line concept and think they have "invented" a game.

In fact, 90 percent of designing is turning a raw concept into something that can be programmed and sold. But if formulating a concept is not tantamount to designing a game, it is a vitally important part of the process.

Truthfully, people seldom get that far with me. I stop the flow of words before they can blurt out that pet notion, since I may already be working on something like it. That can be hard to prove down the road, as has been seen in several much-publicized lawsuits arising from similar situations.

Worse, what if someone says something that ultimately leads to a project? That would certainly merit gratitude, maybe even a mention in the docs. But after six months proposing, selling, writing, and revising, I doubt that I'd want to split the money with someone who'd suggested a vague concept.

Hard as it is to believe after playing poorer products, every electronic game starts with a concept and a theme. The designer may provide the theme, or the publisher may impose one. Whatever the source, the theme is the seed from which the designer germinates the game.

A few projects, by their nature, mandate a particular theme. When a publisher signs a crew to design and develop a boxing game, that's what it expects to get, so the theme is intrinsic to the assignment. The designer may have some latitude concerning the approach to the subject, which is why one company does *Super Punch Out* and another does *Riddick Bowe Boxing*.

Projects for which the designer is allowed to provide the theme are a special delight. But this joy is not tantamount to having the final say over everything. Getting from concept to a final design spec is a lengthy and difficult task. Along the way, the designer will have to make dozens of compromises and adjustments as other team members add their ideas to the main concept.

Games based on original themes are the exceptions in these days of mega-licenses and sophisticated marketing departments.

Today, most new designers only get to provide their own theme after writing several successful license-oriented games for a publisher. More numerous by far are projects tied to a well-known intellectual property. Licenses are the main source of themes, especially for cartridges, so the theme may already be in place by the time the designer joins the team.

That doesn't mean that the publisher totally usurps the designer's role. A publisher often buys a name, likeness, or intellectual property for a game with no thought of how a game is to be made out of it. A license is not a theme per se, and it is usually the designer's responsibility to develop one that exploits the license to best advantage.

All licenses don't offer the designer the same scope. Some impose very strict design limitations. Movie licenses are frequently very circumscribed, since the contract usually requires adherence to the movie's script. TV series and comic books are less restrictive because it doesn't make sense to key a game to one episode out of one hundred or two hundred possible storylines. The characters, images, and background are what the game must capture, and that leaves it up to the designer to invent an interactive context.

Unfortunately, such licenses usually come equipped with a hefty "bible" that explicitly delineates what characters may and may not do. A game that permanently kills Spiderman is sure to run into a storm at Marvel Comics, and don't expect the producers of the television sitcom *Seinfeld* to smile about a plot in which Elaine has Kramer's baby.

Organization and celebrity licenses offer the broadest canvas for innovation. A lot of sports games use them to lend authenticity. Some games put the celebrity's style and attitude into the game, like *Pele Soccer* (Accolade), which has his signature bicycle kick and emphasizes pinpoint passing. But it's hard to see how Ken Griffey Jr., Riddick Bowe, Cal Ripken, or Pat Reilly influence the games that carry their names—much is left to the designer's imagination.

Finally, the designer must stay objective about the theme and not let his or her ego grow too attached to it. This may prove especially hard when it's the designer's own idea. The entire design/development team must ratify the theme or people will waste

valuable time tugging this way and that until the game is in jeopardy of coming apart at the seams.

Professional judgment of themes is ruthless. It has to be because the stakes are so high. Publishers can't afford to fund projects based on unappetizing or uninteresting ideas.

Like cancer, design mistakes are best caught and treated early. The Kunkel Rule of Game Concepts weeds out bad game themes before showing them to the publisher. It says: Fear any game if the premise can't be stated in twenty-five words or less. Even the most complex games must be reducible to a clear statement. Publishers should have that stitched into samplers and hung on the wall of every design department. A nebulous concept inevitably leads to an awkward game.

The Preplay Sequence: Best Foot Forward

There's no true division between the introduction and set-up screens and the rest of the game. Designers handle it separately because it's convenient to write the specifications that way. A full opening incorporates so many elements that it doesn't exactly fit in any neat pigeonhole, yet it's so crucial that it's worth considering as a discreet entity.

The specs for the preplay sequence must describe every step from boot to game-start in minute detail. Most game-writers separate information according to static screen or animated sequence. They fully discuss one segment at a time and describe everything the player sees and hears and the consequences of every possible user action.

The opening contains three segments: the introduction, the credits, and the options. In the past, many games combined all three on a single static screen, but the trend is toward much more ambitious curtain-raisers.

Introductory sequences are noninteractive scene-setters for the ensuing interactive experience. The best produce an ambiance, a context, for everything to follow and promote the player's willing suspension of disbelief.

Removing player interaction cuts programming variables to the absolute minimum. An image in an interactive scene must be viewable from several angles and it must visually respond to a spectrum of user inputs. A noninteractive image can be more

tightly controlled, and hence it uses less memory. The program can allocate the additional resources and memory to make the canned image more intricate and colorful. That's why some games begin with introductions that outshine anything that occurs during the game. An intro is more like shooting a movie; nothing happens that isn't in the script.

A noninteractive introduction can be too ambitious. The current vogue for telling a complex back-story (plot developments prior to the start of the game) in a short movie can slip into overkill. Noninteractive elements rapidly wear thin. They're exactly the same every time they run, so a couple of viewings is enough to perceive every nuance. That's why producers insist on a routine for skipping the preamble, and critics penalize a title a full letter grade if that feature is omitted. A player watches the awesome opening once, maybe twice, and shows it to a few friends. Then it's burned out, victim of the "skip" button.

Titles and credits are the proverbial no-brainer. Between contractual obligations and the publisher's house style, there's not much for the designer to do.

The option screens allow the designer more leeway. They are often a mix of off-the-shelf systems and new ones the designer writes to fit the game's needs.

The Interface: Forging the Link

The system of controls with which the player operates the game is the user interface. In video gaming's infancy, both hardware and software were so simple that no one worried much about command control. In *Space Invaders*, a joystick or paddle moves the cannon from left to right across the bottom of the screen, and the button fires a shot at the incoming aliens. It's hard to imagine a more efficient and intuitive interface than that.

The complexity of modern electronic games poses a considerably greater challenge to the interface crafter. The aim is to keep the control scheme as uncomplicated as possible and still handle all the needed game functions.

Arcade-action games experience fewer interface problems, apart from those caused by spotty programming. Action games generally have fewer subroutines and options, so there's less strain on the control device. One factor that is more important in action

games than in other types is the control system's degree of forgiveness. It's generally not good to require precise positioning to perform activities like going through doors, climbing ladders, and leaping across chasms, but a too-lenient interface may make players feel the game is haphazard and undependable.

It's a delicate balance. If a game tolerates no deviation from optimal execution, it may take a player several tries to perform the simplest actions. Extra latitude reduces frustration but may compromise the game's difficulty.

The design specifications usually make little mention of this aspect. The question is seldom addressed until the game enters the alpha stage, the point at which there's enough of a program to begin in-house play-testing. About the most the designer can do is include a paragraph in the product overview about the target audience that indicates the desired skill setting. The bottom line: This facet of interface construction is in the laps of the gods, which is to say, the publisher's corps of play-testers.

The character of the user interface is far more important in simulators, strategy contests, sports, and adventure/RPGs. All have many more game-functions than any shooter or platform game, so command control is a pivotal issue. More than one flight simulator has flopped because the designer tried to make the player work a joystick and fire button—and punch letters on the keyboard simultaneously in real time.

The keyboard is the boon and bane of computer software. It opens the door for tons of additional functions, impossible with a video game controller, but too many designers abuse the opportunity. Few things are more daunting to the typical gamer than a list of sixty or eighty keystroke commands. Admittedly, this is what some people want in a game, but they don't represent the gaming mainstream of the mass market.

Fortunately, the trend is definitely away from typing. Scott Adams' early text adventure titles used two-word typed commands. Infocom then increased the amount of typing in text adventures, but enhanced the program's ability to understand user orders in the process. Sierra's *King's Quest* introduced a combination of joystick-triggered movement and typed commands. Then came the menu-based interfaces from Sierra, LucasArts, and Dynamix. The latest generation of adventures employs point-and-click icons with an intelligent cursor (one which changes its ap-

pearance to reflect the current action) and on-screen "hot spots" (places that, when touched, cause something to happen).

The Play-Mechanic: The Fun of Doing

The actions, or series of actions, the player performs to operate the game is the play-mechanic. It is also known as the routine-of-play or, in real-time programs, the play-action. It is the process of using the interface to achieve the victory conditions.

The most basic decision about the play-mechanic is whether it should be in real time or turn-based. This is the fundamental difference between *Doom* (id Software) and *Wizardry* (Sir-tech). Both are fantasy RPGs with a combat focus and first-person perspective, but *Wizardry* is turn-based while *Doom* unfolds at arcade speed.

The choice is often made for the designer as part of the commission. Some publishers lean hard toward arcade games, so everything pitched to them should have kinetic elements. Other companies like the extra depth and strategy possible when the player is given time to make decisions between turns.

The subject matter may guide the designer to the right choice when an option exists. A pure statistical sports simulation has no need of action elements, since the object is to manage the real-life athletes. Flight simulators try to replicate an activity that is continuous in real life, so games must proceed in real time or they won't feel right.

In action games, the play-mechanic boils down to a series of controller movements with enough variations to keep the user from falling asleep halfway through the seventh level. The fighting games like *Way of the Warrior* (Universal Interactive) and *Super Street Fighter* (Capcom) are this definition incarnate.

Almost all street-fighting games share substantially similar play-mechanics, no matter what publisher's name is on the box. The courts have ruled, in cases such as the recent Data East-Capcom litigation, that such similarities are intrinsic to this type of game and thus not copyrightable. The important differences between fighting games are found in setting, characters, art style, music, and special martial arts moves.

The play-mechanic is the same across all fighting games not because designers are so dimwitted that they can't think up new

ways to deliver a spin kick. Players of fighting games want a degree of familiarity with the play-mechanic. Having spent hours honing their mastery of this system, they are reluctant to start from scratch with a new control set-up. While fighting games are the extreme, this is almost an axiom for all games: A play-mechanic with familiar elements encourages players because it's easier to learn.

Sound: Gaming's Siren Call

Audio is no longer the stepchild of software development. The advent of affordably priced boards like Sound Blaster II and of multimedia with its CD sound have brought it to new prominence.

The designer's role is limited with sound, unless he or she has the chops to double as composer/musician. The standards in today's games make that extremely unlikely. The specs can suggest an approach and convey the desired atmosphere to the music specialist. At minimum, the designer proposes a musical style (funk, punk, metal, jazz, etc.). Far better to leave the actual music to today's sonic superstars like Tommy Tallarico, the Fat Man, and Marc Miller.

The designer does more if the publisher favors well-known tunes over original ones. Then it's necessary for the specs to recommend pieces for each game-situation. Like all implementation, the music itself is beyond the bounds of the designer's responsibility—and this book.

It's easier for designers to suggest what sound effects are needed, and the specs should list all the events that trigger sounds. Sound effects give weight and solidity to on-screen occurrences. Other team members are always a big help in this area because no one ever thinks up all of them until preliminary play-testing discloses omissions.

Graphics: The Look of Gaming

No element of an electronic game makes a stronger visceral impact than the graphics. Ask someone about a video game, and graphics are invariably the first thing mentioned. Good graphics generate more sales, especially impulse buys, than any other factor, with the possible exception of a truly irresistible license.

American designers put more emphasis on graphics than their counterparts in other parts of the world. It goes back to the encompassing interactive experience philosophy and the conviction that visual stimuli have tremendous power to suck the player into the game.

Even a designer who can't draw has a profound effect on graphics because the specs define the perspective. The designer, rather than the artist, makes that choice because the viewpoint must harmonize with the play-mechanics. The producer makes the final call, but the designer is the main team member involved in reaching the decision.

Utilities: Glad to Be of Service

The least glamorous design task is figuring out the special features and utilities. Sports sim gamers expect instant replays; fans of first-person shooters expect a modem connection so they can play against others via a network, the Internet, an on-line service, or via a direct link. "Necessary" is the adjective that springs to mind. In the long run, such touches separate the A titles from the Bs.

Some publishers want uniform utilities. They supply the boilerplate, which the designer adapts to the requirement of the specific product. It's easy to look at such ready-mades as manna from heaven when the deadline approaches.

Wise designers carefully review this material because a small improvement here can eliminate major points of user-dissatisfaction. Companies aren't so conformist that they won't accept a few beneficial changes. And even the reviewers who add a few oh-so-critical points to a game's rating may not consciously realize why they're doing it. Games with high-class utilities and service features have fewer annoyances to disrupt the willing suspension of disbelief.

Difficulty: Calibrating the Challenge

The skill needed to play a game should correlate with the abilities of the target audience, and most games need to allow for the increased proficiency of players. The design specifications present the structure for adjustable difficulty. This includes both an in-

terface and option screen for the players to adjust the difficulty themselves and guidelines for the programmers on how they should vary the difficulty in the game itself.

The most clear-cut example of a mistake when setting the skill level is a kid's game that has on-screen vocabulary too advanced for its pre- or early readers. More subtle problems are the action game that arcade aces can rip through in an hour, like *Crash 'n Burn* (Crystal Dynamics), or adventures that confront players with a nearly insoluble puzzle before they get into the game-world and develop some momentum.

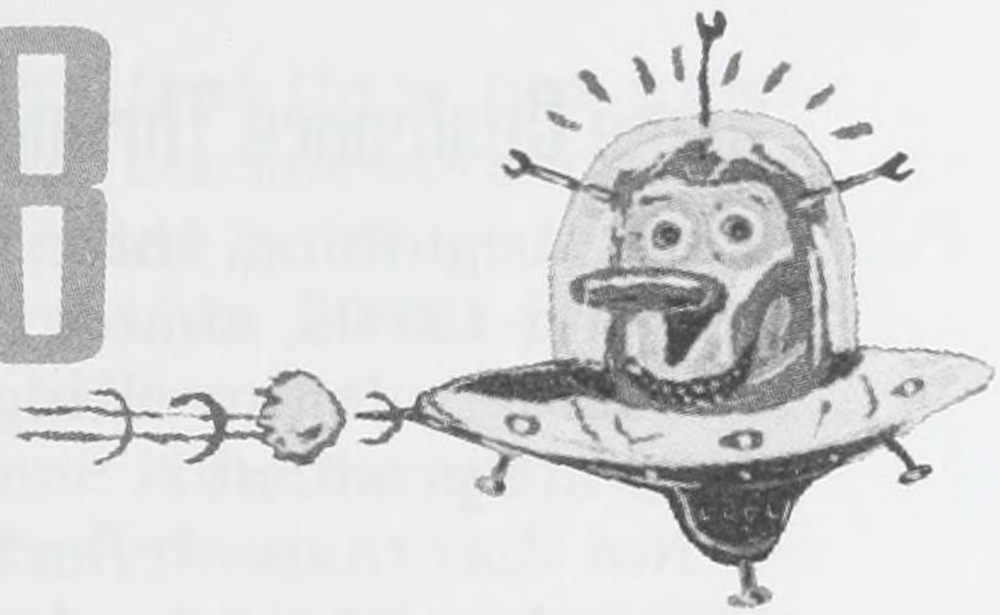
Variable difficulty is the answer to most such problems. Speed is sometimes a good way to vary difficulty, but it's overused. Just speeding up play is the least creative way to tackle the problem. Far better to toughen enemies, increase the number of hazards, and lessen the effectiveness of protections than just wind the game up a little faster.

Sometimes, designers say things like, "It really gets good on level five." That's a sign of a game in trouble. It is not reasonable to expect a player to slog forward too long without getting to the good part.

One solution for helping new players master a complicated game is to provide practice modes. Sega's *Dynamite Headdy* offers three practice modes to help players master its unusual play-action without suffering hours of failure in the game. The *Wing Commander* series (Origin Systems) and *X-Wing* (LucasArts) are mission-based games that let pilots go to school before sending them out to face a hostile universe.

However, it's not easy to figure out ahead of time the right balance between accessible and challenging play and how to best increase the difficulty. Fine-tuning after implementation is always necessary.

Chapter 8



The Game Designer's Role

This chapter is a job description. As always, the requirements are a little different at each publisher and development house, but this is a detailed summary of the activities and obligations of the typical game author.

Today, nearly every game is produced by a team of professionals. A well-organized team can often achieve goals out of reach for any one of its members, but three factors must be in place before a team can benefit from creative synergy: Each member of the team must understand their responsibilities; each member must respect the abilities of the others; and each member must produce their allotment of work.

The second and third points should go without saying—the designer must do his or her work and treat coworkers properly.

The first point is another issue. To be effective, the game designer must understand the designer's role and how that responsibility connects to the rest of the team. Otherwise, there is sure to be conflict. The designer will step on someone else's toes or fail to live up to expectations. With a sense of cooperation and a knowledge of your role, you won't join the world of talented ex-designers who spent all their time wrestling with producers and other implementation specialists until they were out of a job.

Game Designers Through Space and Time

In the beginning, there were no game designers. That is, during the mid-1970s, almost no one had the specific and exclusive responsibility for inventing the content of interactive entertainment software products.

Not that there weren't game designs. It's just that publishers and independent developers didn't feel obliged to pay anyone for "just" specifying game content, which at the time was not very complex. Game specs were usually a couple of pages of vaguely worded promises or even a verbal understanding between the publisher and the developer.

Mostly, it wasn't necessary to make room in the budget for a designer because so many programmers volunteered to do the job for free. Heck, some of them even wrote the documentation, playtested the finished game, and packed the cartons for shipment.

The 1970s and early 1980s were entrepreneurial days. Companies were often small circles of friends with a shared dream. Job titles and descriptions stayed fairly fluid. The idea was to hold down expenses so that the company's owner-employees could share the largest possible profit.

Besides, the quality of the finished product was so dependent on the programmer that game designers had trouble adding meaningful input. The memory of those early machines didn't allow for elaborate designs, and content largely depended on the programmer's ability to get as much as possible out of the hardware. The code crunchers jealously guarded their turf, and they didn't appreciate nontechies horning in on the show.

This way of producing games started to disappear when Trip Hawkins' Electronic Arts chalked up huge sales. EA used teams instead of individuals to produce their games, and they really gave birth to the specialty of game designer.

Hawkins' great insight was that more powerful hardware mandated more sophisticated software. Once Electronic Arts divided game development into specialties, the need for expertise in content became obvious. EA's reputation for quality was earned during those early years, when for the first time specialists could maximize their skills by virtually ignoring everything they didn't do well. The trend has continued, until today it's remarkable when a game *isn't* designed by a team.

Baseball offers a good analogy (except that there has never been a game designer strike). The catcher becomes proficient at playing the position by ignoring skills needed by other members of the team. For example, when Dodgers' Mike Piazza practices he doesn't spend hours fielding ground balls or making the pivot at second base because his practice time is better spent going after foul pop-ups and making the pick-off throw.

The rise of Japan as a world software power in the late 1980s further emphasized production teams, since game development in that nation is a cooperative effort.

No trend is monolithic. A few irredeemable companies with a near-religious devotion to doing everything on the cheap will let anyone associated with the project provide the design, as long as it doesn't cost extra. Outside such bastions of ignorance, no one seriously questions the value of, and need for, a skilled game designer.

Today, the projects are even bigger, and even more planning is needed to keep everyone on track. The game designer is the one who writes the plan that guides the entire project. Nothing gives producers more gray hair than discovering that three artists have spent a week polishing an animation sequence that is no longer connectable to the rest of the game because the programmers have made on-the-fly "improvements" in the original concept. Everyone is supposed to follow the plan the designer has written, and no changes can be made without consulting that person, who is responsible for incorporating those changes into the design document and informing the other people working on the project about it.

The earlier a game is in its development cycle, the cheaper and easier it is to fix. Overhauling incompatible graphics costs nothing if it's done right after the preliminary design document. Wait until the alpha stage, however, and it might add four months to the development period. Postpone still longer, until the prepublication Beta stage, and it's about the same as starting from scratch.

Publishers who once let game auteurs start projects with a one-page synopsis and a brief meeting to sign the contract now insist on more thorough preparation. With so many people working on one game, a designer is needed to provide the overall plan, and often to see that it is carried out as desired.

Prerequisites for the Job

No university grants PhDs in game design, and any apprenticeship program is strictly ad hoc. Chapter Fourteen, "Breaking into Game Design," explains how to get a toehold in the industry, but there are things a would-be designer can do to prepare for a game design career.

Everything starts with a love of, and familiarity with, electronic games. The first commandment of game design is study the field. A designer must possess an encyclopedic knowledge of past and present games. He or she must know what has been tried before and what the results were of those attempts. Without these invaluable reference points, the author runs the risk of reinventing the wheel. The agony over wasted time going down blind alleys is only outweighed by the embarrassment of a publisher picking apart a supposedly new idea that is really old hat.

On the positive side, knowledge of the games increases the designer's speed and efficiency. There's nothing wrong with getting an idea from an existing game, as long as too many elements don't come from the same source. (Designers of *Tetris* clones and *Street Fighter* knock-offs may hold a less idealistic view.) Comprehensive game knowledge also acquaints the designer with valuable negative examples. Seeing something done wrong in another game may keep the author from including it in a new one.

The idea that game designers must be demon players is ridiculous. How quickly someone punches and kicks through *Eternal Champions II* (Sega/Sega CD) has no impact on the ability to create a new game. Analytical ability is far more useful than a quick trigger finger. Someone with no arcade skills will have a hard time designing a challenging action game, but nimble fingers are readily available for minimum wage or free games.

Views about how much the designer needs to know about technical matters run the gamut. An informed layman's knowledge of computer technology is a minimum standard. Knowing how to program doesn't help directly, and many designers lack this training, but anything that fosters understanding of implementation helps the designer do the job.

Keeping up with popular culture is part of the designer's continuing self-education program, too. Popular culture tends to shape the gaming market as well as provide game themes, so the

ivory tower designer is apt to have trouble creating games that push people's hot buttons.

Game Project Organization

There's no single, universal way to organize an electronic game design/development team. Every publisher develops its unique approach. Some companies use complex systems of multiproject interlocking teams, while others just lock a bunch of specialists in a room until they deliver something commercial.

The following is an outline of the way many, if not all, game projects are structured. It's an explanation of the designer's connection to other gamemakers, not a template for starting your own company.

At the top of the heap is the producer. Some companies divide creative and financial responsibilities between two people, but it's more usual to combine those functions.

The game producer (who may have a different title) is like a movie director. Everyone working on the project reports to the producer, who generally has final authority. The producer's power derives from one simple fact: he or she controls the pursestrings. No one gets milestone payments, so important for morale and lunch money, until the producer is satisfied with the work.

Like most people in analogous team-leadership positions, such as professional football coaches, producers are the industry's rovers. Apart from those who also hold big blocks of company stock, producers rarely stay at a publisher for more than five years. Many change jobs almost annually.

Performance pressure sends them packing. In the software business, where projects so frequently go sour, you're only as secure as your last hit. A couple of near-misses, a sudden change in corporate direction, and the producer is desktopping a résumé.

If the producer is not also the product manager, that's the next highest rung on the corporate ladder at most companies. Product managers hang onto their jobs a lot longer than most producers, which indicates cunning if not intelligence. Those godlike personages are so far above humble designers (and book authors) that it may be unseemly to discuss them further.

The game design team is arranged in departments consisting of one or more people. The most important are Design, Program-

ming, Art, and Sound. Each department may be sectioned into subgroups if there are enough people involved to warrant the extra tier of organization.

And, yes, there are some design/development teams with two or three people who divvy up the work among themselves. Id Software, the shareware superteam that did *Wolfenstein 3D* and *Doom*, is a very compact crew led by John Romero and John Carmack. Still, id does only one new game a year, and it's adding design and implementation personnel as rapidly as possible to fully exploit the creative potential of its top game-meisters.

Small, aggressive publishers like id Software sometimes prefer less-structured teams because they deemphasize bureaucracy and the office politics that go with it. This is known as "organization without tiers."

The lead programmer codes most of what gets on the screen, but specialists in noninteractive sequences and 3-D animation are routinely employed on large projects. Publishers like Origin, with its *Wing Commander* series, have departments filled with people who do nothing but vehicle modeling.

Some companies have programmers who concentrate on the frills and/or the utilities. Their mastery of this phase of game development enables them to race through work that would take a general programmer two or three times longer. It's customary for them to participate in several projects simultaneously.

The art director draws almost everything on small projects, but may purely supervise and edit on the most complicated ones. Many companies have a cadre of illustrators who design characters and draw title pages, and 3-D modeling has also become a key specialty in the last five years.

More and more, sound is turning into two allied specialties. Some audio mavens excel in the composition and programming of music, while others have mastered sound effects. There's so much to do in both spheres that it requires at least two people to do the job properly.

They Pay You to Do This?

Rates vary, but publishers do, indeed, compensate designers for work that many diehard game fans would pay for the privilege of doing.

The designer creates an electronic game's intellectual content. For products that don't depend on a license, this starts with the basic concept. For an independent designer, this means he or she comes up with a great game idea and writes a proposal for it. For software publishers, game ideas are often created in company brainstorming sessions, and the designer synthesizes these ideas into a formal proposal. The proposal sets the whole development process in motion.

If the product already has a license, the theme is set in stone. Here the designer's job is to use the license as the basis for an interactive game or simulation.

After a publisher accepts the proposal and commissions a project (in a quaint ceremony known as "sending the check"), the designer buckles down. He or she writes a preliminary design document and/or the final design specifications. A preliminary design is often necessary with complicated games so that the producer can evaluate all of the proposed aspects of the game design and anticipate any problems. The designer takes all of the feedback from this review and incorporates it into the final specification document, which explains every aspect of the game in minute detail. With truly complex games, several designers may be assigned to write different parts of the final specs. The final specifications document can be as hefty as a Stephen King novel, and even for the simplest cartridge game it is seldom less than fifty to seventy-five pages.

Once the producer accepts the final design specifications—a lovely moment because it triggers a milestone payment to the designer—the main part of the designer's job is done. Some companies cut the designer out of the loop at this juncture, much as a movie producer might send the scenarist home when shooting starts.

More often, the designer sticks around in an advisory capacity, helping the producer evaluate implementation milestones and making small adjustments to the final specifications as programmers and artists discover missing pieces or stumble across other glitches. The designer may examine the fruits of the project at several milestone points, from the first graphic demo to the playable Beta version. Occasionally, a design needs additional sections or even massive revisions, which the designer must provide. Virtually all designers have lived that nightmare, since even

the most brilliant specifications document can be sabotaged by unforeseen circumstances.

Unless the contract has a clause to the contrary, the designer is ordinarily not obliged to either play-test or write the documentation. It makes sense for the designer to ensure that the game plays without glitches and that the rulebook is accurate, but other people should do the actual work unless there's additional compensation.

Most designers would rather limit their participation in a game's implementation phases. They would rather write specs for the next game than documentation for the last one.

The Designer Meets the Team

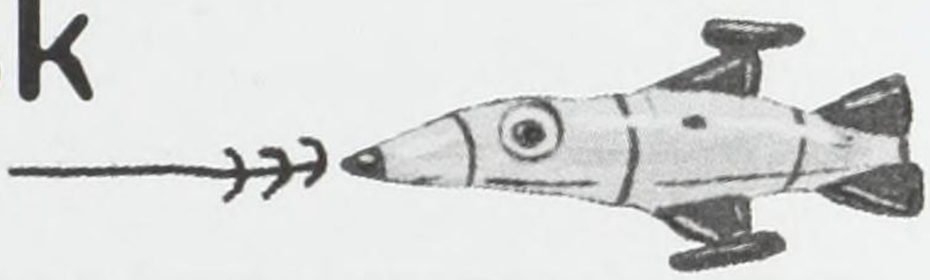
The ability to interact harmoniously and effectively with the other members of the team is nearly as crucial as a knack for creating interfaces. While designers sometimes have only a limited relationship with the development team—if they have been hired only to write the design specs and that's all—more often they accompany a game through the entire process of development in some capacity. The one person all designers will invariably need to develop a good working relationship with is the producer. Some excellent designers are bagging groceries and frying burgers because they didn't take interpersonal relations seriously enough.

The designer must adapt to the inevitable give-and-take among team members. Even more difficult is taking criticism, not always constructive or well-phrased, from people who probably know less about game design than you do. The designer deserves the respect of teammates, but it's a two-way street. It doesn't matter who has a good idea as long as someone does.

The best advice is to listen attentively, filter all input through your own sensibility, and incorporate as much as seems fitting. Sometimes, this won't be enough, like those trying times when you and the producer don't see eye-to-eye. Then it's a matter of keeping the big picture firmly in mind and conceding small points to gain larger aims.

PART THREE

From Dream to Disk

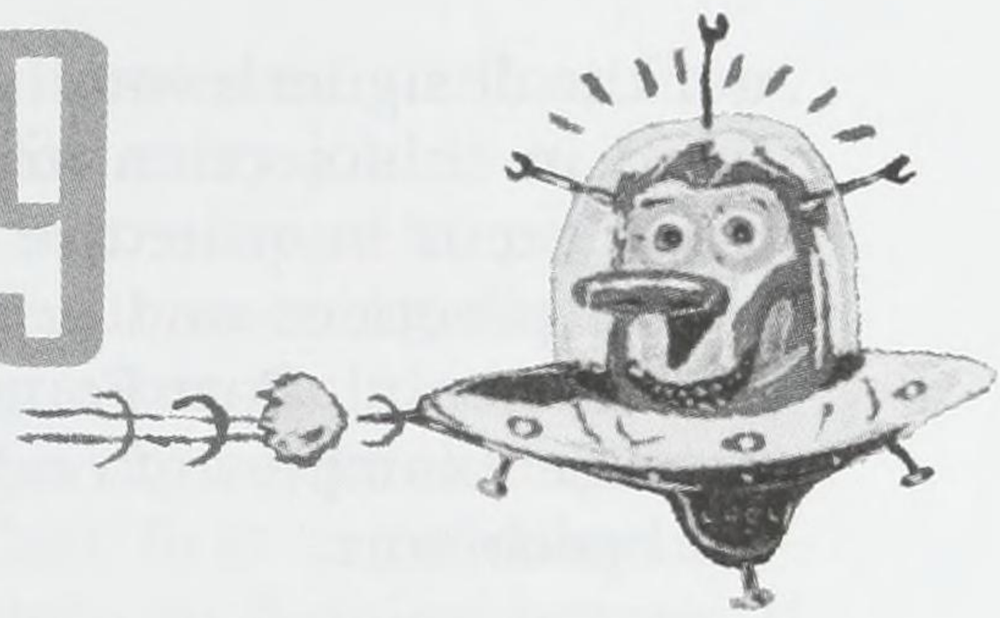


The process of game design is as much craft as art. Flair and insight are vital, but they are not worth much without mastery of technique. A baseball player may have speed, strength, and agility in abundance, but he won't play in the big leagues unless he learns how to judge the strike zone, control the bat, field a grounder, and run the bases. Electronic game designers can't ignore the artistry of their calling, but knowing how to write a proposal or flesh out a specification document is just as important.

Each chapter in this section represents one step along the rocky road from pure concept to a commercially distributed entertainment software product. It gives prospective designers the necessary "how-to" information needed to create a complete game design.

This isn't the only way the process can flow. It's a reasonably accurate map, allowing for each publisher's idiosyncratic procedural variations. Be prepared to be flexible, and never hesitate to tailor your methods to meet a publisher's requirements.

Chapter 9



Sometimes a Great Notion

Before the first line of code, or even a formal proposal, comes the raw concept. It shapes everything that follows. Develop a rock-solid premise, and you are on your way to a good, if not a great, game. Discovering that simple but mysterious bit of inspiration is one of the arts of the designer.

Publishers sometimes flout this axiom, thinking they can simply wrap a game around a newly discovered technological trick or a compelling animated sequence. Those elements are important, but it's impossible to create a satisfying interactive experience without a firm concept at the center. Almost invariably, these games take six months too long to finish, chew up gobs of resources in dead-end development, and disappoint critics and players.

So, you may well ask, where *do* all those good game ideas come from?

The answer is different for every designer. I'm sure my friend Chris Crawford (author of *Eastern Front*, *Balance of Power*, and *Trust and Betrayal*) probably won't recognize his job from the descriptions in this section because his approach to game invention diverges markedly from mine.

Also, video gaming is outside this model to some extent. Mostly, the people who concoct the content for video games are still the same ones who do the programming. Once past the concept stage, many video game companies go to storyboards. From there, the game is programmed and then rigorously tested and tuned un-

der the designer's watchful eye. The stages of game-creation outlined in this section are still more or less the same, but they don't occur in quite the same way in the cartridge universe.

The practices and procedures described here are typical, but not universal. Part Four, "Meet the Designers," has many more specific examples of real-world design inspirations, procedures, and problems.

The Cosmic Light Bulb . . . and Other Fallacies

People always overrate inspiration. It is at the core of the artistic component of design, yet native creativity must be augmented with skill and perseverance. It is the difference between an innate gift for running fast and the finely honed technique of the star hurdler.

Inspiration plays as much of a role in game design as in other artistic endeavors. The designer's genius is the ability to seize whatever inspiration bestows and elaborate these special insights into a coherent artistic creation.

But authors can't wait for the lightning bolt to strike if they want to work, and eat, regularly. If creating games depended on megadoses of pure artistic inspiration, retail shelves would be empty. *Under a Killing Moon* (Access Software/CD-ROM) benefits from many flashes of artistic brilliance, but nothing this immense sprang whole into the mind of creator Chris Jones. Rather, Jones worked on the game in a professional manner, and when things clicked into place just right, he got a fresh idea that invigorated the game.

However, these creative leaps can be elusive, and how much nicer it would be if, just like the cartoons, all we needed was to hold a light bulb over our heads to inspire great ideas. No one has yet discovered the chromium switch that sets imagination in motion, but the artist is not completely helpless. Though inspiration's arrival is not predictable, it *is* possible to encourage it. The intelligent designer sets up the most favorable environment. Athletes call it "getting into the zone." Different artists require different conditions, but figuring out your own is a fundamental lesson every beginning artist must learn. It may be as simple as taking a walk in the woods, listening to a certain type

of music, or, like Rick Gush who is interviewed in Part Four, buying a new pencil and type of paper for every job.

The first and most important lesson is to always be prepared. Inspiration often strikes randomly, without warning. Nothing is worse than getting caught off-guard by a terrific idea. Few authors escape the experience of having a conceptual leap at an inopportune time. It is so clear in that first crystalline moment that it seems burned into the brain in flaming letters. It isn't. And few things are as dispiriting as finding it gone like a sunset shadow before it is fully and irrevocably internalized. Little wonder that so many writers carry notebooks and keep pads on the nightstand.

In "Betty Boop" cartoons the winsome one's grandfather has a foolproof system for instigating inspiration. He sits down and thinks "real hard." You can tell that's what he's doing because he sits in a pose similar to Rodin's "The Thinker." Pretty soon, the light bulb appears over the codger's head, and Betty is well on her way to rescue from her latest predicament.

Inspiring Inspiration

How is it possible to encourage inspiration? Where do designers get ideas for interactive electronic games?

"From life" may not sound like an enlightening response at first, but it's very close to the mark. Experts tell aspiring novelists to "write what you know." That advice is equally relevant to the game designer.

Ivory tower creativity reeks romantic charm. The image is compelling: One majestic intellect fighting a lonely battle to bring forth the Great American Electronic Game. Can't you hear the thunder and see the lightning zigzag across the gray-and-purple sky?

Unfortunately, working alone, in intellectual solitude, is impractical if not ineffective. Splendid isolation may occasionally spark a viable game. Mystics have wandered into the desert in search of illumination since pre-Biblical times, so there is probably something to the power of isolated retreats.

However, divorce from all inputs is an unlikely prelude to a long and productive *career*. That takes staying power, the ability to bang out a succession of interactive titles. Those who fully

understand the consumer market are the ones most likely to produce consistent hits. Human interaction, not alienation, is the road to remaining connected to the shifting social landscape.

No one can tell in advance which inputs are the most valuable. A movie can ignite one imagination and leave another untouched. Even more likely is that a single stimulus will affect each designer in a slightly different way. One designer looks at a traffic-choked expressway and starts to work on a rush-hour road race. Someone else sees the same thing and invents a game about a chicken crossing a busy highway.

A designer never knows when he or she will strike gold. Still, there are a couple of especially productive areas worth special concentration: popular culture and electronic gaming.

Popular culture is indispensable to the designer for four reasons. Paramount is that interactive electronic entertainment is a part of it. Like other forms of entertainment, gaming must capture the mass market to succeed. By studying how other forms of entertainment use pop culture, the designer may pick up valuable hints on how to do it, too.

Second, a pop culture artist must know the target audience. Seeing what makes them respond in other media generates lots of helpful data. Our society's movies, TV, radio, comic books, and other mass media mirror our collective desires and concerns. High culture gets the space in galleries and museums, not to mention most of the official sanction, but popular culture is a fiber optic connection to the mass mind of America. Games that push popular culture's hot buttons have the best chance to grab enough people to support commercial publication.

Third, pop culture is visceral and immediate, and its vibrant images evoke powerful responses. More people can identify a picture of Mario or Mickey Mouse than can pick out the secretary of state in a photograph of the president's cabinet. The identification and use of these images is a key skill for any designer.

Finally, pop culture is also worth studying because it's the source of most licenses. Software publishers don't acquire an intellectual property unless it is recognizable to millions. A designer who is familiar, at least in a general way, with a mass media property can produce a more effective proposal and endow the finished design with the essence of that license.

You may wonder where this leaves fine art (painting, literature, poetry, dance, and such). The fine arts are ennobling, and in the largest sense, they nourish the artist's soul. Alas for culture vultures, they don't yield a fraction of the licenses that derive from more plebeian artistic expressions.

Fine art often requires a cultivated sensibility. One learns to appreciate the nuances of classical music and abstract expressionism. Pop culture is bare knuckles art that smashes into our brains with hurricane force. For every person who enjoys the music of Mahalia Jackson, a thousand buy the records of Michael Jackson.

The other thing good designers do is study the field itself. That sounds paradoxical after the earlier injunction against stealing from the arcades, but it's not. A gamemaker who takes all the concepts for a new title from an existing one is a thief; one who adopts elements from many diverse games is just plain intelligent.

Studying other games is not just a good idea, it's a necessity. Some beginning designers become so obsessed with trying to be "original" that they prefer to start from square one in every phase of the game. This is a liability. Wasting time to reinvent the basics for every facet of a design puts one additional obstacle across the path of a good electronic game. It also makes it unlikely that the author can write the necessary material within the industry's notoriously tight schedules.

Originality is precious, but making a game different for the sake of difference is counterproductive. Through trial and error, designers have solved many nagging little problems, and it's foolhardy to snub such accumulated wisdom.

Sometimes the best approaches are so intelligent and appropriate that they become codified as design conventions, and respecting these conventions can make sense. If players are used to the *Street Fighter II* control system, new fighting games can draw on the established audience by laying out order entry with the same philosophy. An experienced designer knows the conventions and manipulates the players' expectations to convey the feeling that the game is very intuitive. If you would be unconventional to put extra spark into a design, choose wisely.

Studying games also provides the designer with instructive negative examples. Knowing what doesn't work can be as impor-

tant as understanding what does. Seeing something that was botched in another game may inspire the designer to do it better.

Finally, as a designer you need to know the market. What competitive games have done sets the standard for what any new entry in a category must deliver. Ignorance of alternative products makes it hard to build an "A" title.

The results of not studying the field can be ludicrous as well as painful to the pocketbook. One software company president used to buttonhole Bill Kunkel and I annually at an industry convention. The conversation always went the same; it was scripted just as certainly as Hulk Hogan's next main event.

"We've really got the goods this time," he would bellow at us confidently. Then he'd drag us to a private viewing room.

"Wait till you see my new games," he would chortle. "They'll knock your socks off!"

We were both adverse to going sockless and less-than-tactfully skeptical about the probability of such a startling result. Our evident lack of runaway enthusiasm drove the executive to greater oratorical heights. "I'll bet my balls they'll knock your socks off!"

They never did. (We didn't make him pay up, though.) The games always looked about four years out of date, and it didn't surprise me to learn that the company president didn't even have a computer and never looked at other publishers' offerings. And it shouldn't surprise you to learn that the company eventually departed the field.

Narrowing the Focus

The process of turning an idea into a game concept is a bit like lapidary. A gem cutter takes a dull, rough nugget of sapphire and chips away at it till there are a hundred smooth facets that twinkle brightly.

However the game author arrives at a theme for a new game, it usually needs some additional work before it can be shown to a publisher. There's nothing sadder than a quick turn-down on a potentially great idea because it's inadequately or improperly explained. There's rarely a second chance to propose the same idea.

To help illustrate the process of turning a raw idea into a proposal, and of turning the proposal into a preliminary and then a

final specifications document, we will track the progress of a hypothetical CD-ROM called *Highballin': Electronic Model Railroading.*

The idea comes from my childhood experiences with Lionel and HO-gauge trains. I love building and operating layouts, but my ardor cooled as a child when dreams of a magnificent rail empire collided with space limitations. I could construct a small set-up, a few circles of track with rudimentary scenery, but that modest layout, though fun, never allowed for the full scale of the operations I imagined.

Never having had enough room to indulge this hobby, I've had no model railroad for many years. It struck me that other railfans might be in the same predicament.

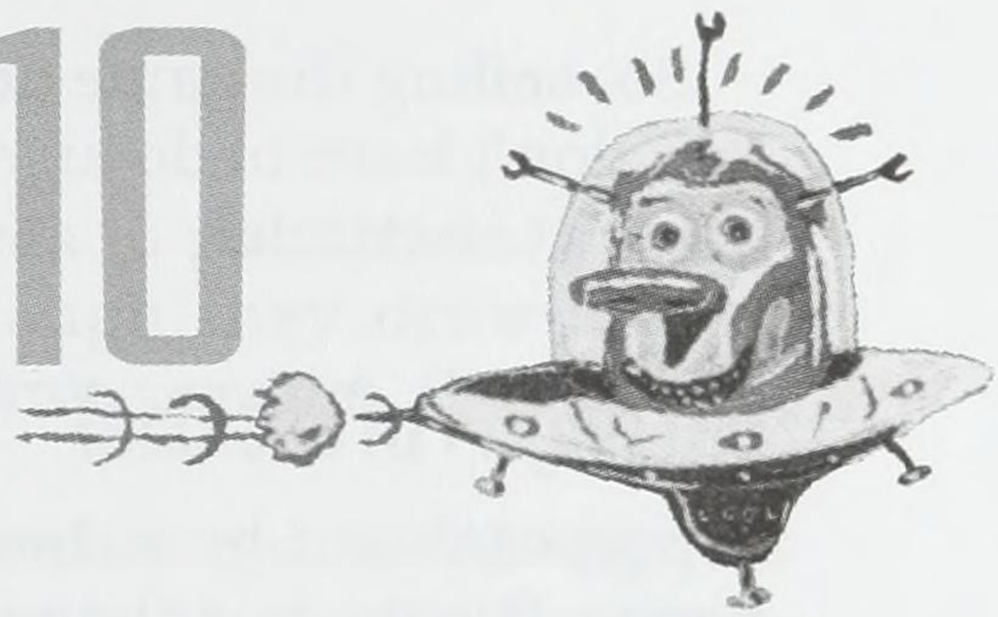
With the subject firmly in mind, I needed to find a name for the new game. *Highballin': Electronic Model Railroading* is descriptive of my intent, and it can serve as the working title. It gives a handy label by which to refer to the project, even if the final name on the box is something entirely different.

In homage to the Kunkel Rule of Game Concepts, my next step is to write a short statement of purpose for the proposed project. Here's what I tossed into my game-idea file:

Highballin': Electronic Model Railroading is an interactive multimedia model railroad game and activity program. One or more participants assemble freight and passenger trains, make pick-ups and deliveries at fully animated sites, and if desired, operate the railroad on a machine-generated timetable.

Although this is more than twenty-five words, it is less than fifty, and it contains the essential tenets of the design. It says that *Highballin'* is a multimedia product with multiplayer capability, that its theme is model railroading, and that it is an entertainment product with some game elements. Now that I have the idea clearly stated, I can begin to turn the idea into a proposal.

Chapter 10



The Proposal

In one sense, the proposal is the designer's most important work, since it represents the first incarnation of the proposed game. Like Dr. Frankenstein, the designer must give this newborn creature life. Monster movies call it the "divine spark." For games people, the life catalyst is a publisher's contract and binder check.

It amounts to the same thing. If a proposal doesn't get a good grade from a publisher, it lies there inert and unliving, a collection of cobbled together parts. If the proposal doesn't achieve its objective, the backing of a publisher, anything else you might learn from this book becomes purely academic. If the proposal doesn't sell, the project may be over before it starts. While it is possible for a development house to revive a project by producing a demo based on the design, this is generally beyond the means and capabilities of a new designer.

Sometimes the designer gets to present an idea face-to-face. It's a great opportunity for the designer who thinks fast and has the gift of gab. Getting instantaneous feedback from the person who spends the publisher's money is a potent stimulant to creativity.

In-person presentations aren't standard procedure. Sometimes it's a courtesy extended primarily to proven hit-makers. Even when the product development manager grants an audience, it's likely that no commitment will be made. The best a designer can reasonably expect is a hearty handshake and a promise to give the written proposal a thoughtful and thorough review.

So, selling the game depends on a convincing written proposal. You don't have to do anything very fancy with the format, but the content is crucial.

Getting Professional

A proposal can be submitted in pencil on the back of used shelf paper. It can, but the chances that anyone at the receiving end would read that mess are almost nil. If you want to have the best chance of getting your proposal accepted, it needs to be polished, well-written, and attractive—in short, professional. Maybe it shouldn't be that way, but it is. There will always be stories about unusual pitches that succeed. The reason they're told and retold is their rarity.

A prospective designer is like any budding writer. Talented candidates keep submitting work until they make a sale, but the number of hopefuls dwarfs the short list of those who get a contract. And while the constant influx of new electronic game designers testifies that it is possible to turn pro, there's a wide gulf between the possibility and the start. Jumping that gap, selling that first game design, is always a struggle.

The numbers are not encouraging. Major publishers get at least a hundred unsolicited game proposals a month. A publisher that introduces twelve games a year, exclusive of SKUs, is pumping a lot of product onto the market. Some well-known companies market half that number or less.

That sets the odds at literally a hundred-to-one against getting through the screening process. Even though some of those hundred proposals will be amateurish and easily dismissable, many will be serious submissions in good form. With that kind of competition, why make your proposal look any less than its best?

A little investment in the right equipment is money well spent. A computer with good word processing and spellchecking software and a laser printer have become the tools of the trade. A modem is an asset once the project is underway, but it's not a necessity at the beginning.

The Agent: Threat or Menace?

About the time they start to submit ideas to publishers, designers think about hiring an agent. Though they draw a solid fee, plus expenses, agents provide many services that benefit the professional.

An agent removes the burden of negotiating deals. Designers who cut their own contracts are saddled with conflicting agendas. As artists, they wax enthusiastic, but a negotiator must remain cool, aloof. The more strongly the designer commits to the project, the more likely the publisher's representative is to lowball the price. "We really want you to design this game," they will say as the crocodile tears spill down their ruddy cheeks. "If only you could take a little less money to fit within our budget." Too often, the designer discards financial interests in favor of the pleasure of working on a pet project.

Letting an agent hassle with the publisher also smooths the subsequent working relationship. A designer who successfully hammers a producer for an extra 1 percent royalty may have alienated someone with whom he or she must work harmoniously for the next twelve to eighteen months. Far better to let the agent fight the battles and draw the wrath.

Most development houses already have representatives, or an officer of the company, who handles the business. The independent designer is in the same position as any other new writer; most agents don't want to handle a client who hasn't sold a game. Some freelancers negotiate for themselves, but most people who depend on design work for income have a representative.

The best way to find an agent is by talking to established designers. Keep a short list of possibles, agents who might be sympathetic to your work. If a publisher indicates that it is seriously interested in your proposal, most agents will be willing to step into the picture and add you to the client list.

The Laundry List

Sometimes there's a proposal that precedes what is normally meant by a proposal. It's a document that briefly explains many different raw concepts. Such a report, usually produced at the publisher's behest, contains few of the elements normally in-

cluded in a game design proposal. It's an ice-breaker that focuses mutual attention on specific project ideas.

That's no small accomplishment. Exploratory meetings between designers and product managers and/or producers can sound like outtakes from Marty. Everybody smiles and says things like, "I dunno, what do you want to do?" until everyone gets tired and goes home.

The publisher's reasons for requesting a laundry list are varied. It shows what kind of projects most excite the designer, which some publishers consider important. The designer's degree of enthusiasm may mean the difference between an inspired design and a slick, professional job. There isn't a publisher who wouldn't rather have the former because the best games come from people who are passionate.

A laundry list may help a publisher clarify its own marketing strategy. Even successful companies like Electronic Arts periodically reach points at which there is no sure path ahead. They hope that a catalog of game ideas will incite productive discussion.

Small publishers sometimes want laundry lists from designers because they can't afford to fund anything less than a surefire hit. The lists let them review a couple of dozen candidates in less time than it takes to read through the average full-length game proposal. Laundry lists let hit-hungry companies cast the widest possible net. The sheer variety is appealing. Who can resist a buffet?

Opinion is sharply divided about game idea lists. Some designers regard it as a foot in the door, a chance to catch a publisher in a receptive mood. They don't mind tossing out a lot of ideas because it doesn't take long to spin a new concept and write a paragraph about it.

Others worry about the fate of those ostensibly rejected concepts. They fear that one of those ideas will percolate in some executive's brain and reemerge as someone else's project.

Another reason to dislike laundry lists is that they may offer publishers too many choices. Some companies may react like a kid in a candy story. Ideas compete for attention, and the publisher can't settle on one.

The designer's artistic aspirations may be a point against game lists, too. When an author develops a dozen game themes in less than a week, he or she can't be equally close to all of them. Many

want to focus all their salesmanship on one or two ideas and not risk getting caught on a project that grew out of a theme they put on the list mostly to fill out the document.

Despite the objections, newcomers should ponder carefully the opportunity to submit a catalog of game concepts if a publisher asks for one. There's wasted brainstorming, and ideas do occasionally, inadvertently slip through the cracks, but it can be a shortcut to getting a project funded.

Making a List

Unless the publisher mandates a specific number of entries, eight to twelve is a good number. A dozen ideas is plenty of variety, but the overall document won't end up so lengthy that it exhausts the reader. A document of this type, which seldom runs more than three or four pages, doesn't overwork the designer, either. That subdues worry about the fact that a concept document is written on speculation. (Translation: it's a freebie.)

Never lose sight of the concept document's purpose. It must spark interest, or the publisher won't continue to the next stage, the full-dress proposal. Each entry must not only capsulize the essence of a game but sell it, too.

Take a hint from the best advertising copy. Concentrate on the big picture, not the little details. The game-theme must come ahead of interesting but minor details about the interface. Most publishers, when they look at a short proposal, are thinking about commercial viability. Stress the aspects of the project likely to attract consumers and give less weight to things that won't increase its salability.

This is how I'd position *Highballin': Electronic Model Railroading* in a laundry list report:

> Everyone loves electric trains, but who has the space for a dream layout? *Highballin': Electronic Model Railroading* solves the problem by bringing this beloved hobby to the mass audience.
>
> *Highballin'* lets one or more armchair engineers assemble freight and passenger trains, make pickups and deliveries at fully animated sites, and if desired, run the railroad on a timetable. Multimedia audiovisuals, including smooth digitized

video, give *Highballin': Electronic Model Railroading* excitement and realism far beyond any simple tabletop train set. *Highballin'* also offers the possibility of add-on disks with extra scenery, rolling stock, and engines.

Note the use of the present tense. Some proposals are written in the future or a conditional tense, and I've always thought that produced the wrong impression. Far better to talk about *Highballin'* like it's a real product. That will make it seem more solid to the reader. It's a small matter, but never overlook any possible edge.

The Proposal: An Overview

Laundry lists, as indicated, are an exception to common industry practice. Far more usual is a request for a written proposal about a specific game idea.

This four-to-six page document generally determines the fate of any game idea. Despite an understandable aversion to working without compensation, most designers put an inordinate amount of time into polishing the proposal.

No template exists for the perfect design proposal. Each designer arrives at a format that fits his or her style—and then tailors it to meet the special needs of each job. Any proposal format that is well-written and logically organized, and that sells and explains the game, fills the bill.

Some general guidelines are possible. The document should be single spaced, with skipped lines between major headings to set them off from other sections. (I like to skip a line for every new paragraph, but that's an idiosyncrasy.)

Top, bottom, and side margins should be generous, an inch all the way around. That makes it easy for the reader to scrawl margin notes for later reference. This should always be encouraged because such interaction may subtly draw the reader into the project.

Type specifications depend on individual taste. Let readability be the guide. A twelve-point serif typeface is a safe choice because so many people have vision limitations. Bookman, Times, and Courier are three good fonts.

Put a copyright notice on every document, as well as a notice that it is confidential. It is not necessary to file a copyright notice with the government. The intention to copyright the material is sufficient. Another mandatory element, forgotten more often than seems possible, is the submitter's name, address, and phone number. If the publisher doesn't know who you are, how can they fill out the advance check properly?

A Proposal Outline

The sample proposal below consists of eleven separate headings. Each tells the publisher's representative about a specific phase of the game, and however you organize your own proposal, each of the game aspects listed below should be included. There's no room for detail or complexity; the proposal sells the sizzle and outlines what the designer intends to do in the full design specifications document.

Following is an explanation of the eleven sections of a written proposal. For an example of a complete proposal, see Appendix A: "A Sample Proposal."

I. Introduction

The opening section is like the advertising trailer for a new movie. The objective is to hook the reader so he or she will continue further into the document instead of chucking it into the return/reject pile. Use strong statements with action verbs.

The introduction should communicate the writer's enthusiasm for the theme as well as suggest why people would buy an interactive electronic entertainment product of this type.

II. Product Overview

This is the linchpin of the whole proposal. It's where the author explains, in broad strokes, the nature of the game or entertainment product.

The overview usually starts with a brief statement of the game's theme and purpose. Some designers consider this so vital that they break it out as a separate heading, labeled "Mission" or "Theme." The overview must also cover the target platform(s), number of players, support for network and/or modem play, and the character of the interactive experience. A video game pro-

posal also estimates the size of the cartridge in megabits and whether the cart incorporates battery back-up on systems in which this is available.

Conspicuous by its absence in most proposals is any reference to schedule. People have grown skeptical about optimistic timelines. This, like the money, should be worked out with an agent acting as intermediary.

III. Opening Sequence

Designers once left a description of how a product opens until the final specifications. There's such a premium on audiovisual trimmings that it's wise to include a rough version of the curtain-raiser in the proposal.

While the introduction should be technologically feasible, it is more important to convey the way it will strike users than the intricate mechanics of its construction. The designer writes this section as if reporting the opening portion of a movie or play, concentrating on what the gamer sees and hears.

A designer already affiliated with a development house would also have a color drawing or even a sample on disk to illustrate the opening. The independent designer usually has to make due with an especially vivid description.

IV. Graphics

Under this heading is a general discussion of the game's art, animation, and imagery. A drawing of the main display screen is a valuable supplement to the written version. Designers who can't draw and don't know a friendly artist should at least sketch a schematic of the playfield to show the proportion of the screen devoted to the active display and any on-screen controls and readouts.

Other aspects of graphics included in this section are whether the main graphics are drawings or video, the perspective used for each play-screen, and any special visual effects. The author must describe the visual style of the game and identify any distinctive graphic elements that will distinguish the game from its rivals.

V. Sound

This heading is sometimes combined with graphics under the heading "Audiovisuals." They are most apt to be treated together

when the designer isn't part of an existing development team that includes sonic specialists. However, any multimedia proposal must discuss the product's audio component.

It's hard for a nonexpert to say very much about sound, but designers can set an aural ambience for the game. This segment of the proposal references the type of music, sound effects, and speech to be incorporated into the product. If sound is a game-element, as in *Ecco the Dolphin* (Sega/Genesis), this should be detailed under a separate "Sound" section.

VI. Play-Action

The program's play-modes are the subject of this section. Whereas the overview describes the user's ultimate objectives, play-action covers what the player does, in game-terms, to achieve victory conditions or to utilize available activities.

The space given to the interface should be proportional to its novelty and sophistication.

VII. Play-Mechanics

This heading refers to how the user manipulates the program to produce the desired results. It's a point-by-point description of the manner in which the user interacts with the program.

The control system should be described with some specificity in this portion of the proposal. Producers, who evaluate designs at most companies, always want to know as much as possible about this facet of the potential title.

VIII. Difficulty, Playability, and Play-Life

It's far too early in the project for the designer to have definitive comments about difficulty and play-life. Despite uncertainty, though, the designer must define the degree of challenge and explain how that level can be adjusted to meet the needs of users with varying amounts of skill.

The proposal must estimate how many hours of entertainment the finished product will deliver. If the play-life is "unlimited" or "infinite," this should be stated, along with reasons, because this is a particularly desirable quality. That's a snap if the game is open-ended, like a military simulation or a sports title, but it may not be as easy to judge a shooter or adventure.

IX. Special Features

There's no room, in a document of this size, to give much detail about more than two or three high-visibility features. The designer should anticipate the aspects of the game that will make good selling points in ads and on the back of the package.

X. Market Perspective

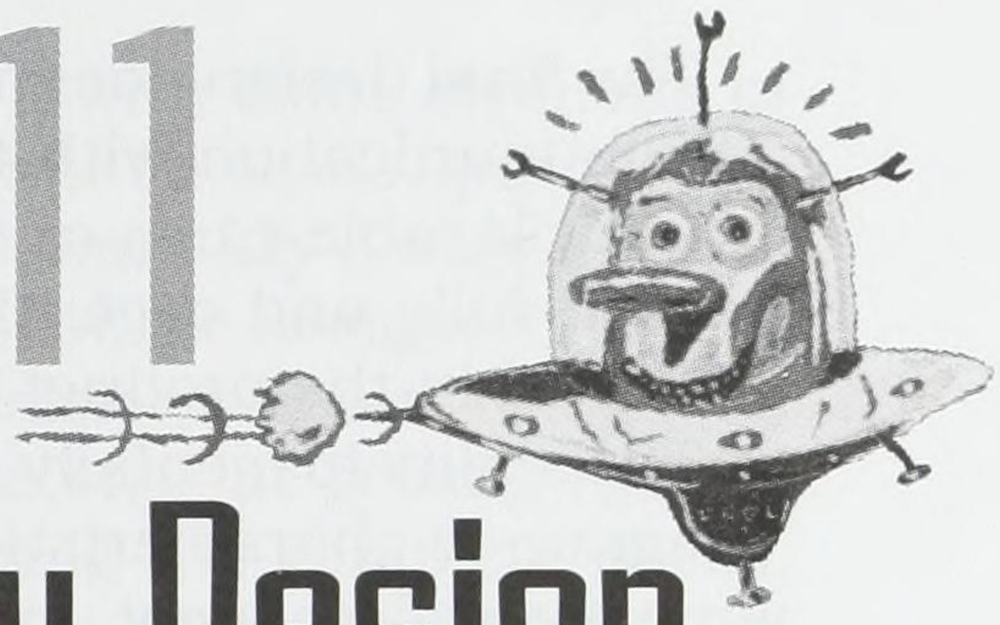
Most designers don't like to hear about track records and market trends, but they're an inescapable part of the business. This section pays homage to that reality by positioning the proposed product within the electronic gaming field.

One way to enhance a proposal's appeal is to cite other games of a similar theme or format that have sold well. A favorite tactic is to compare the proposed product to a similar, successful one—and then point out the ways in which the new entry would be superior.

XI. Summary

The wrap-up section usually harks back to the introduction with a couple of general statements about the suitability and salability of the proposed product. It is also a good spot to describe the designer's expected role in the project, if it gets the green light.

Chapter 11



The Preliminary Design

The real work on a new game begins when the proposal turns into a project. Up to that point, the designer is more a sales representative than a creative artist. Some designers have enjoyed short-term success based solely on their knack for salesmanship.

Once the publisher bites, and sends the enabling binder check, hype goes out the window. The focus shifts to the product itself, and memos about delivery dates replace glib quips about the glorious concept.

The shift from potential to performance is embodied in the specifications document. Vague promises are replaced by minutely described game-elements and play-systems. The designer must validate the proposal's claims with unambiguous "how-to" instructions for the implementation team.

The final design demands as much sustained creativity as a novel. The designer has two months, or less, to write a consistent and rigorous set of instructions. It is the ultimate test of the designer's ability.

The specs have the same relation to the game as a playwright's script has to a mounted production. Actors may ad-lib dialog, and the director may change the stage directions, but the script remains the reference point. Developers may stray from the specs, but the producer is always there to make sure that the game comes together according to the intent of the overall plan.

The final design document is the cyber-author's main medium of communication with the people who actually turn the design into a playable game or entertainment. If something isn't in that report, fully and cogently described, the chances that it will be included in the product are greatly reduced.

The ability to mentally play through a game before it's a bootable program is characteristic of the finest designers. The most sublime mental concept, no matter how vivid, is just an inconsequential daydream until it is communicated faithfully to the implementation specialists.

Prelim Specs: Pro and Con

Designers are no more industrious than other creative artists, so extra work is not popular. Most would prefer to go straight from the proposal to the final design specs and call it a day. Preliminary specs can seem like just another sheaf of paperwork that stands between you and the main event, your finished game.

Sometimes a preliminary write-up is a necessity. It can even end up saving time. A preliminary report solidifies the consensus about the project and thus speeds up the work. It brings issues to a head and may prevent later rewrites.

This intermediate step is especially useful when the publisher is unsure about what the game should be. That may sound unlikely to those who have never worked on a game, but it happens frequently.

Games based on big-ticket licenses are prone to such indecision. Caught up in a frenzy of competition, the publisher buys the rights to an alleged can't-miss property without thinking about how to incorporate it into an interactive entertainment product. It is only after the licensing contract is signed that the publisher starts to think about how to turn the newly acquired prize into a good disk or cartridge. And frequently, the only way to get maximum punch out of many licenses is to have a product on the shelves yesterday, before the allegedly hot property suddenly goes cold.

Since the licensing negotiations invariably last a month longer than anticipated, the project starts on a hurry-up schedule. This leads to panicky product conferences in which people say things

like, "It should be a little like *Harpoon II* with some elements of *ESPN Baseball*, and could you work in a little *Sonic Spinball*?"

In general, and in cases like this, a preliminary design specifications document does three key things for the designer: First, it forces the publisher to make basic decisions about the game. Second, it gives the designer one more shot at getting helpful feedback before buckling down to writing the complete specs. Last, it establishes a paper trail proving that the designer acted professionally and in good faith throughout the project. Sad to say, there's plenty of finger-pointing when a development project spins out of control, and careful documentation can provide reputation-saving evidence.

The Preliminary Specifications Document

The preliminary design should be no more than twenty to twenty-five manuscript pages, roughly three times the wordage of the original proposal. Because excess effort is always to be avoided, the easiest way to organize the preliminary design specifications is to build on the foundation of the proposal. Done this way, it is essentially an expansion of the document that closed the deal in the first place. This report provides specifics, adjusted according to any comments the producer made about the proposal.

The designer must never forget the purpose of this extra submission: to force the producer to define the design parameters as narrowly as possible. Every issue that remains vague is a potential source of discord later in the development cycle. The designer should never be afraid to make choices because each one compels the producer to state a position. It's as important to know what the producer doesn't want as what he or she does.

When there are two or more legitimate ways to execute the same phase of a game, the designer can pose the alternatives as a direct question to the producer, giving the pros and cons of each. The temptation to orchestrate such multiple choice questions so as to lead the producer to a desired choice isn't playing fair. Much better to make a clear-cut recommendation, if the designer has one to make, and treat all options objectively.

This section, drawn from a hypothetical preliminary design spec for *Highballin': Electronic Model Railroading* shows how a designer would handle one of these knotty questions:

IV. Graphics

There are four graphic modes used in *Highballin'*:
D. Animated Sequences
1. Perspective
When a player drops off or picks up cargo or passengers at a station, depot, or siding, a close-up, angled side perspective replaces the running view of the train. Instead of showing the whole train moving down the track, the view encompasses only the length of one car, with the cars ahead and behind it in the train only partially displayed.
2. Loading/Unloading
When the view zooms in tight on the loading/unloading site, the appropriate noninteractive sequence runs. Each site has at least two sequences, one for loading and the other for unloading. The final design specifications will contain full scripts for every noninteractive loading/unloading sequence.
Question: Should the loading/unloading scenes be artwork or full-motion video (FMV)?
There is merit in both choices, though the designer recommends rendered, bitmapped artwork. Factors to consider include:
a. Artwork is cheaper and easier to produce than FMV.
b. Artwork is more flexible. Small changes in background artwork and adding and subtracting bitmapped objects from the scene will allow the same basic artwork to be used for several different sites without producing the degree of sameness than would result from repeated use of the same FMV footage for different sites in the game.
c. FMV will produce a more realistic effect and lend a degree of authenticity impossible in a tabletop layout.
d. The use of artwork seems acceptable for a product that simulates model railroading because it would allow us to preserve some of that fantasy-daydream ambience of toy trains.

The designer has taken a definite stand in favor of artwork over FMV while recognizing the argument in favor of FMV. By asking for a decision, the designer insures that he or she won't have to guess about such a crucial matter.

An Outline for Preliminary Specs

Following is an explanation of the eleven sections of a preliminary design document.

I. Introduction

With the product already sold, the designer throttles back on the "sell." A good approach is to edit the corresponding section of the proposal for brevity. For *Highballin'*, the introduction might read:

> I. Introduction
>
> Remember model trains? Just about every adult does. Sad to say, that's just what it must be for most of us, a memory. Fewer and fewer families have room for their own model railroads.
>
> *Highballin': Electronic Model Railroading* puts a full, working tabletop railroad within the reach of everyone's pocketbook. People can enjoy the challenge and satisfaction of running a rail empire without worrying about where to put the equipment—or how to pay for it.

II. Product Overview

In a preliminary design, the product overview is primarily for the convenience of the people associated with the project who are not part of the design/development team. Executives in the marketing, sales, advertising, and promotion departments need information about the product. They may not have the time or insight to wade through the specs. The designer must encapsulate the game and its salient features in this section, even though its points are restated in greater depth later in the report.

II. Product Overview

Highballin': Electronic Model Railroading is an activity toy with game elements that simulate tabletop model railroading. The solitaire user can design a layout, create a scenic background, assemble and break up trains in the yard, and run a timetable operation.

A strong emphasis on audiovisual effects conveys the "feel" of running an accessory-laden *Highballin'* train set. This includes authentic train sounds, smoking locomotives, animated loading and unloading sequences, functional track signals, and much more.

Highballin': Electronic Model Railroading is intended for CD-ROM and Windows CD. Editions for 3DO and Saturn are under consideration. The product is expected to appeal to older consumers who have nostalgia for trains and teenage boys who haven't had the chance to fully enjoy the hobby.

III. Opening Sequence

Full detail is appropriate for this section. Most producers want the graphics started early in the development cycle because that's the game-element that takes longest to generate. A ready-to-implement introductory section is a sizable asset.

A rewrite of the proposal text serves as the general description, but it must be supplemented by a step-by-step run-through. This excerpt, again from Highballin', demonstrates the technique.

III. Opening Sequence

General Description

When the disk boots, the user sees an antique-looking model train catalog. It opens to a double-page spread of a gadget-laden model train layout.

Train sounds begin as the image zooms into the catalog, inside the frame of the layout picture. It focuses on a train

building up steam at a rural station. The picture morphs from a two-dimensional image to a pseudo-three-dimensional one.

Engine sounds are loud as the train starts to gather steam to pull away from the station. The scene rotates so that the train is coming directly out of the screen at the player. As it roars into the foreground, the title comes up superimposed: *Highballin': Electronic Model Railroading.*

The Sequence

A. The Closed Catalog

1. Graphics

   The designer will provide photos of several actual catalogs so the artist can do a fictitious one in the same spirit.

2. The Animation

   The catalog flips open to the double-page spread.

   Art Question: Should the pages flip open, or should a hand turn them? I like the hand, but either way is acceptable. If the pages turn themselves, some sort of visual effect should surround the catalog.

B. The Open Catalog

1. Graphics

   The spread shows an angled side view of a train in a typical catalog shot. The whole train is visible, though the engine is nearer the foreground.

2. Sound

   The sound starts low and builds steadily.

   a. Steam engine sounds

   b. Clicks of rails and steel wheels

   c. Train whistle—distant and widely spaced at first, closer and louder as the morphing progresses

3. Animation

   a. The screens zooms in tight and the catalog framing disappears off the edges of the screen.

   b. The two-dimensional train morphs into a view of the same train as it will appear when the user is running the railroad in operations mode, except that the vantage point is closer to make the train larger on the screen.

C. The Onrushing Train
   1. Animation
      When the morphing is complete, the display starts to rotate so that the engine faces directly out of the screen toward the user. It scales larger.
   2. Sound
      The train sounds and rail clicks continue, louder and faster. The train whistle gets louder and more sustained.

D. The Logo Screen
   As the train scales larger, before the locomotive disappears, the logo of the game is superimposed over the picture.

E. Credit Screens
   1. Graphics
      At this point, the logo is on the screen, superimposed over the image of the onrushing train. When the engine is as large as practical, it freezes.
   2. Sound
      When the train picture freezes, the sound diminishes to a low level and continues.
   3. Animation
      The credits scroll vertically down the screen, using the publisher's house crediting style. When the last credit runs, the screen fades and is replaced by the first option screen.

IV. Graphics

The designer divides this major heading into entries for each aspect of the graphics. For *Highballin'*, the relevant game-elements are the construction kit screens, the operations mode main display, the loading/unloading sequences, the first-person mode, the train yard mode, and the utility screens.

These descriptions are more effective if combined with thumbnail sketches. Thumbnails are ultra-simple drawings, the kind almost anyone can do. They express the arrangement and on-screen spatial relationships between game-elements, though they aren't usually realistic or accurate renderings of specific images within the sketch. They give the folks who can draw an idea of what the designer has imagined. At this stage in the development process, designers who can't draw (like the author of this

book) can usually con the project's art director into helping produce thumbnails of each display in the game.

In projects where some thumbnails already exist, generated for the proposal, the producer may request another round of drawings. Designers aren't expected to be good enough illustrators to do this. So unless the designer is that gifted individual who can also double as an artist, such drawings are done by other members of the design/development team.

A tentative list of objects, divided by game-mode, is often included in the preliminary design document. This is sometimes combined with a list of animations needed for each on-screen image. This text from the *Highballin'* preliminary design shows one format for the object list:

IV. Graphics

A. Object List

1. Operations Mode
   a. Brakeman
      i. General Description
         The brakeman is a railroad employee, frequently seen at signals, entrances to yards and other similar places.
      ii. Animation
         (a) Swinging lantern (to stop train)
         (b) Throwing a turnout manually
         (c) Shooing away something from the track (works with dog and hobo objects)
         (d) Waving to train

V. Sound

The design must cover sound, including music, speech, and sound effects, under a separate heading in the prelim specs. The presentation parallels the graphics, except that sound samples aren't available yet.

For each object listed under graphics, the sound section should list the collateral speech and sound effects. It is not necessary to script the dialog, but it helps implementers to know that some-

thing will be required. Here's how a designer might handle sounds associated with the brakemen in *Highballin'*.

V. Sound

A. Object List
1. Operations Mode
a. Brakeman
i. General Description
The brakeman's sounds would ordinarily be masked by louder sounds around him. However, some artistic license will be taken with regard to speech. Also, some brakeman actions create audible sounds.
ii. Speech
(a) "Stop!" (while waving lantern)
(b) "Move along there!!" (while shooing)
iii. Sound Effects
(a) Sound of throwing the turnout

VI. Interface

This section contains control systems for every phase of the product. A schematic of each interface used in the game reduces the chance that the programmer will misunderstand the command-control system. In games that have redundant order entry, including keyboard control when mouse or joystick are supported as well, all alternatives should be briefly described.

For each play-mode, the designer lists and explains every order entry device. In the *Highballin'* specs, this section would describe the control systems for utilities, train assembly/disassembly in the yard, the operations mode, and the campaigns (timetable and untimed modes).

VI. Interface

A. The Utility Screens
These include the presession option screens, the Railroad Hall of Fame board, and the scheduling screen in the campaign mode (timetable version).

1. Moving the controller up and down highlights the header of each menu in turn.
2. Pressing the button while a header is lit activates that menu.
3. Moving the controller to the left or right highlights each of the choices within that menu.
4. Pressing the button when a choice is lit confirms that selection.
5. When the user confirms a choice, the header of the next menu automatically lights up.
   Exception: The player types his/her name when prompted. Hitting the 'Return" key enters it and causes the sledder screen to appear.
   a. Keyboard Control of Utility Screens
      i. The Space Bar moves the highlight from heading to heading.
      ii. The Tab key highlights choice to choice within a menu.
      iii. The Return key functions as the button.

VII. Play-Mechanics

The document would be too big if this section gave full routine-of-play descriptions for every mode in the game. For *Highballin'*, that would probably require as many pages as the entire preliminary design report.

Rather than do a spotty job that gives the producer a bad impression, the designer generally saves this job for the final specs. The explanation of the play-modes in the proposal can be repeated here.

VIII. Difficulty, Playability, and Play-Life

If the corresponding portion of the proposal was written correctly, the designer has little to add on the subject for the preliminary design. Most elect to put the effort into other sections and insert a reedited version of the proposal entry.

IX. Special Features

Because there's a separate Construction Kit heading, it is sensible to put everything about that mode, including its graphics, sound, interface, and play-mechanics, in the same section rather than parceling out the data among several headings.

The preliminary design should have subsections on each of these four facets of the construction kit, as well as a summary of its operation. The outline of this section in the *Highballin'* preliminary document would read like this:

IX. The Construction Kit

General Description

The user pulls together the components of his or her rail empire using the overhead-perspective construction kit screen. After picking the natural terrain, the hobbyist lays tracks, positions trackside features, and locates loading/unloading sites on the railroad's right-of-way.

The player chooses engines, cars, and buildings from banks of objects. Data bank categories include: locomotives (diesel and steam), rolling stock, terrain/scenery, people, fauna and flora, railroad and private buildings, signals, and all the other things that make model railroading so entertaining. Most items can be "painted" one of several colors. The user can also decorate locomotives, rolling stocks, and structures with famous railroad logos if desired.

A. Graphics
    1. Perspective
    2. Parts Drawings
    3. Painting and Weathering
    4. Railroad logos
B. Interface
    1. Taking items from data banks
    2. Removing items from the layout
C. The Data Banks
    1. Locomotives
        a. Steam
        b. Electric
        c. Diesel
            i. GP-7
            ii. GP-20
    2. Tank Cars
    3. Refrigerated Box Cars

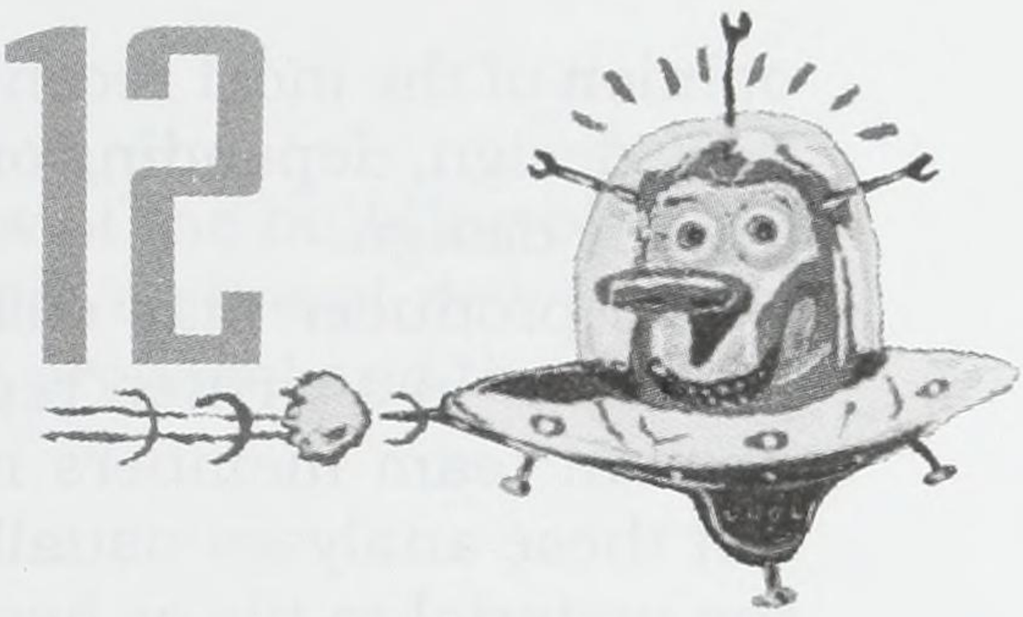
X. Market Perspective

This section should be cut to the bone. If the designer has marketing issues to put on the table, a separate memo is preferable. The implementation specialists are not directly concerned with marketing, and extraneous discussions only elongate the document.

XI. Summary

The designer issues a call to action in the wrap-up section of the preliminary specifications. The standard approach is to refer to major areas that await the producer's decisions and suggest starting and completion dates for the final design specifications.

Chapter 12



Final Design Specifications & Beyond

Ring the bell for the main event. Everything else is only preamble to writing the final design specifications document. All that the designer has accomplished so far has been the set-up for the big show. If the final design isn't on target, the project is in deep trouble.

Even a preliminary design, important as it sometimes is, still leaves room for the designer to sidestep challenges and finesse the vague spots. The twenty-page preliminary report gives the game-author the convenient excuse of limited space. The final specs are as many pages as it takes the designer to cover everything that happens from the moment the user boots the disk or cart until he or she flips the "off" switch.

A finished game design can be as large a volume as this book, so logistics prevent inclusion of a complete one for *Highballin': Electronic Model Railroading*. While such a document might be interesting, an unabridged spec would also be too specifically tailored to one game to serve as a template. This guided tour should prove more helpful. It discusses all the elements of a full design, with expanded examples from *Highballin'*, to steer the prospective designer through the construction process.

Preparing to Write

Few designers are brave enough to start the final design before the producer has had a chance to comment. The project chief's

opinion of the most recent milestone (the proposal or the preliminary design, depending on the project) is the charter for completing the design.

The producer may deliver the verdict in a meeting, over the phone, or in a written report. Personal style varies. Other development team members may also comment on the manuscript, but these analyses usually go directly to the producer, who adds the material to his or her critique.

The designer must be attentive to what the producer says. The producer generally gives the designer a chance to argue points, but it is still a writer-editor relationship. The designer can champion the aspects of the game he or she feels are the most important, but the designer can't afford to be too stiff-necked or egotistical. The producer is the boss, and the designer must eventually become reconciled to the project leader's directives.

The Final Design Format

The exact format of the specifications document is not important. There's no standard structure, nor would there be much gained if one were established. What *does* matter is that the designer must structure the specifications in a logical, easy-to-read format that the implementers can understand.

The most depressing questions from programmers always start, "Where in the specs can I find . . ." Such queries tell the designer that he or she hasn't gotten the instructions across to the people who are realizing the vision.

The length varies. A designer attached to an in-house or independent development studio may write as few as forty pages for an action game. An independent designer might fill two hundred pages with specifications for an adventure or a sports simulation, and some documents run over four hundred pages!

Design docs are growing fatter every year. That reflects the increasing complexity and depth of today's electronic entertainment. A three-hundred-page book should have a table of contents, and an index is a good idea, especially if you can use the automatic indexing feature of some word processing and desktop software.

An Outline for Final Specifications

Following is a section-by-section review of the building blocks of a final design document. Note that there a several new sections that expand on the outline used for the proposal and the preliminary design.

I. Introduction

This report stresses function over style, so designers don't waste a lot of time with sections that don't have a direct bearing on implementation. Copying the introduction from the preliminary design is sufficient. (See Chapter Eleven for guidance in editing down the introduction from the proposal if no preliminary specs exist.)

The only addition is the table of contents. Nothing elaborate is required. The minimum is a list of the major headings and the corresponding page numbers. The best is an outline of the entire document with a page number next to important subsections.

II. The Theme

A succinct statement about the product near the front of the report crystallizes the project. It will also give people outside the development team a comprehensible hook to base advertising, promotion, and carton graphics. The statement might read as follows for *Highballin'*:

> II. The Theme
>
> *Highballin': Electronic Model Railroading* is a solitaire activity toy with game elements that simulate tabletop model railroading. The user can design a layout, create a scenic background, assemble and break up trains in the yard, and run a timetable operation.

III. Product Overview

Like the human appendix, the product overview is vestigial by the time the designer reaches the final design. It is useful for recapping whatever decisions the publisher has made about platforms and on-sale dates, but it is generally only a few short paragraphs.

IV. Opening Sequence

If there's a preliminary design, this section is 90 percent done. If not, the author must expand the proposal entry. It must not only describe every individual element, but it must also tell the director how to combine them into a presentation.

The opening noninteractive sequence for *Highballin'* would be elaborated as follows.

IV. Opening Sequence

General Description

When the disk boots, the user sees an antique-looking model train catalog. It opens to a double-page spread of a gadget-laden model train layout.

Train sounds begin as the image zooms into the catalog, inside the frame of the layout picture. It focuses on a train building up steam at a rural station. The picture morphs from a two-dimensional image to a pseudo-three-dimensional one.

Engine sounds are loud as the train starts to gather steam to pull away from the station. The scene rotates so that the train is coming directly out of the screen at the player. As it roars into the foreground, the title comes up superimposed: *Highballin': Electronic Model Railroading*.

The Sequence

A. The Closed Catalog

1. Graphics

The preliminary sketch of the catalog provided by the art department is excellent, but it would benefit from a few more touches. Recommendations:

a. A more antique look

i. Feathering of page edges

ii. A crease in one of the corners of the cover

b. Paler color palette

The colors are too vibrant when the catalog first appears. A more washed-out look would be more realistic and heighten the contrast later.

2. The Animation
   a. Mists swirl up around the catalog, obscuring the background.
   b. The front cover of the catalog slides open.
   c. The other pages flip over quickly, blurs of color, and the catalog finishes open to the center spread.

B. The Open Catalog
   1. Graphics
      The spread shows an angled side view of a train in a typcal catalog shot. The whole train is visible, though the engine is nearer the foreground.
   2. Sound
      The sound starts low and builds steadily.
      a. Steam engine sounds
      b. Clicks of rails and steel wheels
      c. Train whistle—distant and widely spaced at first, closer and louder as the morphing progresses
   3. Animation
      a. The screen zooms in tight and the catalog framing disappears off the edges of the screen.
      b. The two-dimensional train morphs into a view of the same train as it will appear when the user is running the railroad in operations mode, except that the vantage point is closer to make the train larger on the screen.

C. The Onrushing Train
   1. Animation
      When the morphing is complete, the display starts to rotate so that the engine faces directly out of the screen toward the user.
      a. The engine, now in straight-front perspective, scales larger.
   2. Sound
      a. The train sounds and rail clicks continue, louder and faster.
      b. The train whistle gets louder and more sustained.

D. The Logo Screen
   1. The scaling stops when the locomotive reaches the largest possible size without noticeable pixelization.

2. The background fades to a solid color and everything behind the locomotive goes invisible.
3. The logo appears, superimposed over the picture of the locomotive.

E. Credit Screens

1. Graphics
   When the locomotive freezes and the large logo appears on the screen, the credits start to roll.
2. Sound
   a. The whistle does not blow again until the last credit is on the screen.
   b. The sound of the track clicks and the locomotive itself fall to the level of background noise.
   c. The product's theme song, "I've Been Working on the Railroad," starts to play. This should be an up-tempo version of the standard, perhaps even a Dixieland arrangement.
3. Animation
   a. The credits scroll vertically down the screen.
   b. Entire credit list runs. (It is customary to provide the complete list of credits, and this is often put into an appendix at the back of the specs document. If that has been done, a note to that effect would be added here.)
   c. When the final credit has run, the first utility screen appears.

V. Utility Screens

The screens that let the user set parameters, choose game-modes, and customize the rules generally get separate treatment in the final design. Following is how the inventor of *Highballin'* delineates a screen.

Note the reference to a flow chart under "General Description." Flow charts have become a standard part of every final design package. A simple schematic, showing routes a user can take through the product, reduces confusion and answers numerous questions before they are even asked. It is common to tack this kind of material on the end of the specs document as an appendix.

V. The Utility Screens

General Description

This section describes the screens that offer users presession choices in the approximate order in which they come up on the screen. The exact order and relationship of utility screens is established by Appendix B: The Flow Chart.

A. Mode Select Screen

This screen allows the user to choose which activity he or she will pursue in the play-session about to start.

1. Graphics

The presentation should be simple and uncluttered, with the menus filling most of the screen. An artistic frame with a railroad motif would help establish the ambience. Theelements of the screen are:

a. The game logo (top of screen)
b. Menu items are listed one below the next with the choices within each menu arrayed horizontally.
c. At the bottom of the screen is an on-screen button labeled "Done."

2. Feedback Sound

When the user picks among the choices for any given menu item, the feedback tone sounds each time a different option is highlighted.

3. Menu Items and Choices

Controller:	Joystick	Mouse	Keyboard
Speed:	Slow	Normal	Fast
Game Status:	New	Continue	
Game Mode:	Operation	Construction	Yard

a. Definition of Terms Used on the Option Screen

Controller: The mouse is the preferred method of control, but there is full support for all three.
Speed: How fast things happen during realtime racing. "Slow" is 10 percent slower, while "Fast" is about 10 percent faster than "Normal" speed.
Game Status: *Highballin'* lets the user continue an existing game over more than one play-session.
Game Mode: This item allows the user to pick one of the three main facets of the simulation.

b. Default Settings
Controller: Mouse
Speed: Normal
Game Status: New
Game Mode: Operation

4. "Done" Button

Clicking the "Done" button ratifies all choices and causes the next screen to appear. The following choices trigger the corresponding screens, as follows:

a. New; Operation
   i. This triggers the operation screen.
   ii. A pop-up window prompts the user to select a track layout file. This file can be either one the user has previ ously saved or one of the stock track layouts contained on the game disk.
   iii. When a layout has been selected, a new pop-up window asks the user to select a locomotive file. This file can be either one the user has previously saved or one of the stock layouts contained on the game disk.
   iv. When the user picks a locomotive file, a pop-up window asks the user to pick a rolling stock file.
   v. When the user picks a rolling stock file, a pop-up window asks if the user would like to go to the construction kit to modify the selected files.
      (a) If "Yes," the operations display appears with "Modify" as a default.
      (b) If "No," the operations display appears.

b. New; Construction Kit
This triggers the construction kit screen.

c. New; Yard
This triggers the yard screen.

d. Continue; Operation
   i. This triggers the operations screen.
   ii. A pop-up window prompts the user to select a saved game file.
   iii. When this is done, the file loads and the operation display becomes active.

e. Continue; Construction Kit
f. Continue; Yard

5. Control System
   a. Moving the controller up and down highlights the header of each menu in turn.
   b. Pressing the button while a header is lit activates that menu.
   c. Moving the controller to the left or right highlights each of the choices within that menu.
   d. Pressing the button when a choice is lit confirms that selection.
   e. When the user confirms a choice, it automatically lights the header of the next menu.
      i. Keyboard Control of Option Screen
         (a) The Space Bar moves the highlight from heading to heading.
         (b) The Tab key highlights a choice to choice within a menu.
         (c) The Return key functions as the button.

VI. Main Display(s)

Many programmers prefer to have all aspects of the main display pulled together into one section rather than parceled out among several different headings, like Graphics, Sound, and Interface. It makes it easier to see how the various elements function together.

For programs like *Highballin'*, which have multiple displays, the section explicates each under its own subhead. This is an outline that one author could flesh out to make a full spec.

VI. Main Display Screens

A. Operations Mode Main Display
General Description

The user runs the railroad using the control panel and sees the train move through the track layout.

1. Graphics
   The operation mode main display is composed of the following visual elements:
   a. Time Line
      The following information is reported on a line at the top of the display:
      i. Time: a 24-hour digital clock
      ii. Day: the day of the week
      iii. Timetable/Unscheduled: shows whether the trains are running according to a schedule or if the layout is being operated without reference to this consideration.
      Note: This choice is made on the operation customizing screen, reached through the command strip beneath the status line.
   b. Command Strip
      Clickable buttons perform a variety of functions:
      i. Customize
      Clicking this button leads to the operation customizing screen, on which the user can set parameters for this product mode.
      ii. Schedule
      This button triggers the schedule screen, which shows currently planned pickups and deliveries.
      iii. Map
      This button triggers the map screen, which has an overhead map of the layout territory. On that screen, the user can zoom in on any desired area.
      iv. Financial Statement
      Triggers the financial summary screen, which shows the user how the railroad is doing financially.
      v. Yard
      Triggers the yard mode's main display.
      vi. Construction Kit
      Triggers the main display of the construction kit module.

c. Active Display
This rectangular area fills the entire screen between the command strip and the control console at the bottom of the screen.
d. Control Console
This fills the lower quarter of the screen, edge-to-edge. It has the following components:
i. The Track Schematic
This shows the layout in straight overhead perspective with turnouts indicated by large buttons with arrows to indicate which way the switch is currently thrown.
ii. The Engineer's Cab
The on-screen controls look like a model railroad controller with two speed controllers, two whistles, and two reverse controls.

2. Sound
a. Control Sounds
b. Operation Sounds
(and so on, describing each sound element)

3. Interface

B. Yard Mode Main Display

C. Construction Kit Main Display
(and so on, describing each aspect of each main display in full)

Thumbnail sketches are a must. Every display screen should be pictured, at least in rough, on a separate page. Some designers put the drawings as close as possible to their descriptions in the document, but most group them at the back of the manuscript in the order in which they occur in the text.

VII. Objects

A tentative list of objects, divided by game-mode, lets illustrators focus on producing these items. This is sometimes combined with a list of animations needed for each on-screen image. This text from the *Highballin'* final design shows one format for the object list. However, note that it is highly abbreviated—with large gaps in the numbering—in order to give some idea of the scope and detail required.

VII. Objects

A. Construction Kit Objects
    1. Locomotives
        a. Steam
            i. 4-6-0 Ten Wheel
            ii. 2-6-0 Mogul
            x. 2-8-2 Mikado
            xx. 2-8-4 Berkshire
        b. Diesel
            i. EMD FTA/FTB
            vi. Alco RS1
            x. EMD GPP7
            xv. EMD SD-7
        c. Electric
    2. Rolling Stock
        a. Freight
            i. Ore Car
            ii. Hopper Car
            x. Refrigerated Box Car
            xi. Cattle Car
            xx. Flat Car
            xxi. Flat Car w/ containers
            xxii. Flat Car with automobiles
            xl. Box Car—1930s
            xli. Box Car—High Cube
            l. Bat Window Caboose
        b. Passenger
            i. Standard Pullman
            ii. Standard Sleeper
            viii. Streamlined Pullman
            xii. Streamlined Vista Dome
            xv. Streamlined Baggage Car
    3. Buildings
    4. People
        a. Brakemen
        b. Station Personnel
        c. Water Tower Operator

5. Track
    a. Straight sections
    b. Curved Sections
    c. Turnouts
        i. Left
            Available left turn radiuses:
        ii. Right
        iii. Wye
6. Scenery
    a. Mountains
        Sizes available:
        i. Gentle Rise
        v. Steep Grade
        x. Mountain with tunnel
    b. Streams/Rivers
    c. Trees/Shrubs

B. Yard Mode

C. Operation Mode

A designer who anticipates the needs of implementation specialists expedites the project. Collecting photographs and illustration for the art department for as many objects as possible gives the graphics a big boost. The artists will ask for such materials sooner or later, so accumulating them early in the development process saves time, since illustrators won't have to search for images themselves.

VIII. Animations

Constructed in parallel with the Objects section, this one lists all the items again. This time, though, the writer describes any animations associated with that object.

IX. Graphics

A lot of the visual effects are discussed under other headings, so this is a catch-all for graphics not described elsewhere. The designer provides a written description to complement each display screen thumbnail, using the same format as was used to define each of the utility screens (Section V above).

X. Sound

The word processor's select-copy-paste routine gets another workout in this section, which again runs through the object list. Subheads for each object detail the sound effects and/or speech associated with each object. In games that utilize speech, which is just about all of them these days, the designer often compiles all spoken phrases into a supplementary document.

Why doesn't the designer make one big object list with the graphics, animations, and sound for each? Some do. It's logical, but it's inconvenient for the implementers. Different people work on the sound than work on the graphics and animation, so each specialist appreciates getting pages with nothing but relevant material.

XI. Music

Outstanding talents like Tommy Tallarico, Fat Man, Mark Manning, and Russell Lieblich are interactive entertainment stars, and their specialty has become much more important due to the multimedia boom. Five years ago, a paragraph took care of music in most design documents, but it can't be glossed over like that any more.

The musical requirements are detailed here, too. An original score requires the least effort, and the most ingenuity, from the designer. Few game-authors know enough to communicate musical ideas to a true professional, and doing it on paper is even less likely to work out well. Descriptions of the desired feel and/or intended effect characterize many documents.

If the producer decides to go with public domain music or acquires rights to one or more compositions, the scope of the job shrinks. The designer's task is to pick the right music for each need from the bank of tunes.

XII. Interface

If this document was preceded by a preliminary spec, the interface is already in place. All the designer needs to do is give it a last polish. If the designer jumps directly from the proposal, he or she must handle the interface as described in Chapter Eleven.

XIII. Play-Mechanics

The final design includes routine-of-play descriptions of every phase of the game. These descriptions keep the designer's colleagues mindful of the overall project and may help others spot gaps in the specs before they sabotage the game.

Highballin' has four play modes: Free Operation, Timetable Operation, the Railroad Yard, and the Construction Kit. Since each also has its own heading, the designer tenders little more than an overview. The technique is best explained with an excerpt:

> XIII. Play-Mechanics: An Overview
>
> A. The Construction Kit
> B. The Railroad Yard
> C. Free Operation
> General Description
>
> One of the greatest pleasures of model railroading is watching the trains run and seeing the layout features do their stuff. This play-mode simulates that experience.
>
> 1. Control System Overview
> a. The user employs the controller or keyboard to move a cursor around the on-screen controllers.
> b. When the user points the cursor at an on-screen control, pressing the left button activates it. The activated control is surrounded by a glowing outline to indicate that it is "live."
> c. When an on-screen control is lit, what happens next depends on whether it is a button or a lever.
> i. Levers
> The only levers are the ones that govern speed and the locomotive's whistle or horn.
> (a) Moving the controller forward increases speed (on the throttle) or loudness (for the whistle).
> (b) Moving the controller back has the opposite effect.
> (c) Pressing the left button deselects the controller.
> (d) Pressing the right button at any time brings up a load/save dialog box.
> ii. Buttons
> D. Timetable Operation

XIV. The Construction Kit

In the final specs, every special feature and play-mode needs to be treated separately; in the case of *Highballin'*, this includes the construction kit. This is another section that depends on whether there's a preliminary design or only an accepted proposal. If the former, as in *Highballin'*, the designer expands the outline for the construction kit with full descriptions of every function and system. Otherwise, the designer compiles the section as shown in Chapter Eleven.

XV. The Yard

The railroad yard, where the trains are assembled and broken down, would be described in the same format in which the designer presented the construction kit.

XVI. Timetable Operation

The designer offers the full specifications for this play-mode, originally mentioned among the special features in the proposal. This includes a thumbnail drawing and a routine-of-play description for the balance sheet screen and other secondary displays invoked only in the yard mode.

The designer tries to give the programmer working on this mode everything he or she needs to set it up. Its connections to other game-modes would be on the flow chart.

XVII. Difficulty, Playability, and Play-Life

This facet of the game, as previously mentioned, gets attention during the alpha stage of development. Repeating previous statements on the subject is adequate.

XVIII. Summary

Although it may seem like an invitation to disaster, the designer calls for feedback from the producer, and through the producer, from the rest of the design/development team. Even though this may inspire backseat driving, getting objections and suggestions on the table as soon as possible allows that much more time for necessary revisions. It should be expected that as soon as the final design specifications have been handed in, a slew of questions will follow.

Beyond the Final Design

Putting "30" at the bottom of the last page of the design specifications document is like the final "The End" on a Frankenstein movie. It may seem like everything is over, but the sequel is inevitable.

The design document is the most important milestone, but it doesn't mean that the job is done. That's not until the approved beta copy of the game goes to the disk duplicator or the cartridge manufacturer. Until then, the designer must be available for several kinds of work.

The most important is revisions. Sometimes things don't go as planned, and it's necessary to revamp the plan in light of new knowledge. Designers strive to get it right the first time, but corrected documents close gaps or rework phases that don't pan out during testing.

For example, in *Highballin'*, it's likely that the command strip would be moved early in development and installed just above the control console. This makes it easier for the user to move the cursor among the control elements. A change this small probably wouldn't require an extra report, but the producer might very well ask for a new thumbnail and a memo alerting the rest of the team to the new arrangement.

One of the joys of the development process is its synergy. The product develops momentum, and that sometimes generates some good ideas that weren't obvious at the start. Occasionally, there's even new technology to be considered. Either or both eventualities send the designer back to the keyboard to create supplemental reports. It is rare, but not unheard of, for the designer to go back and create a whole new final design that incorporates such expansions.

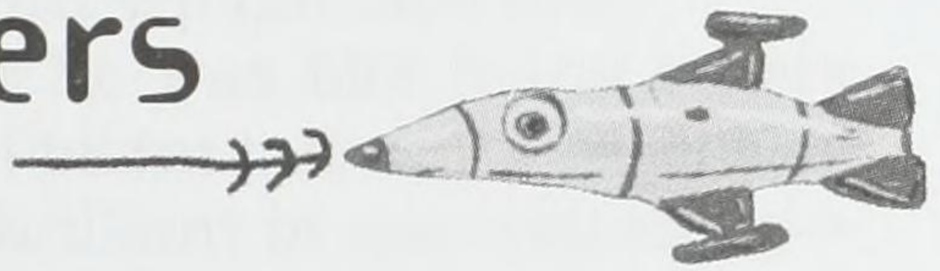
The need for better communication may prompt the producer to ask for more material. If the implementers can't understand the specs as written, they must be rephrased. The designer's vision can only be fully realized when it's fully shared. Most designers try verbal contact first, but if that doesn't solve the comprehension problem, the affected portions of the document itself must be restated.

After implementation starts, designers should expect a flow of questions. As difficulties crop up, the designer employs his or her expertise to help move the team past such obstacles.

And then, finally, comes the blessed day when the designer knows that he or she has helped bring a product all the way to market. That's the day the author first hears the rumor that his or her magnum opus has been in the stores for a week. Then it's just six or seven increasingly whiny phone calls to the producer, and the author can see the finished form of the game or entertainment that began as a twenty-five-word concept a year or more earlier.

PART FOUR

Meet the Designers



There's a big gap between theory and practice. The history, philosophy, and methodology of game design are essential, but none of it describes what it's like to be a designer. In this section, we get a chance to meet the men and women who make their living creating games.

Some may be disappointed to learn that there is no single "game designer lifestyle." Individuality is in plentiful supply among the people interviewed here, as might be expected of any group of artists. For every designer, there is another unique story of how he or she got into the business.

These interviews with the leading game designers of the 1990s display the full range of attitudes, personalities, and approaches in today's game design field. Broad similarities will be apparent, but it is the richness of creative diversity that reigns supreme.

Mike Berlyn

An industry veteran of over fifteen years, Mike Berlyn's game credits are quite extensive and run across a variety of platforms. He has worked on *Oo-topos*, *Cyborg*, *Gold Rush*, *Congo*, *Suspended*, *Infidel*, *Cut-Throats*, *Fooblitzky*, *Pass Times in Tonetown*, *Dr. Dumont's Wild Parti*, *Altered Destiny*, *Snoopy's Game Club*, *Columbo's Mystery Capers*, *Motile*, and *Bubsy* (Accolade/SNES and Genesis).

Additionally, Mike has had four novels published. *The Integrated Man*, *Crystal Phoenix*, *Blight* (under the pen name Mark Saunders) and *The Eternal Enemy* (with William Morrow).

But before Mike started his career he was like many others who became designers: He had a lot of talent but no focus. "I was good at everything—not excellent or brilliant in any one thing—but above average at everything. If I hadn't turned to game design, I don't know what I would have done."

As a result, his route into the business was long and circuitous. "I kind of tripped into it," he admits. "In 1979, I was writing novels and decided to purchase a personal computer to use for word processing. After six months, I discovered the original adventure game, *Colossal Cave*. I just thought, 'Wow! This is really interesting. I should learn how these are done, because it's writing and computers and a lot of fun. So let me figure out how to do them.'

"I set about teaching myself how to program computers, and read all sorts of books. In about six months, I'd written a game, and my wife and I started a software publishing company, called Sentient Software. We published it, and that was really the start of how I began designing games."

One of the high points of Berlyn's design career was *Bubsy* in: *Claws Encounters of the Furred Kind* (Accolade 1993). Not only did the game score with the buying public, but it represented an evolution in Berlyn's creative style. Each level of *Bubsy* was designed as a whole instead of in the usual single-screen blocks. As a result, the game scrolled easily through a contiguous landscape that moved vertically as well as horizontally.

"Typically, I would spend several hours standing over an eight-foot-long strip of graph paper, laying out and actually sketching

out ideas of what this level would look like. I would stare at areas and say, 'Hmm, now a tree would be interesting because Bubsy could jump on top of it, then bounce up to this island floating in the sky.' I'd sketch an island, and my art director would actually make it look like something good. My art designer and I worked side by side: I would sketch and lay out the look and the feel of the level, and she would draw it in behind me. This long, laborious, and painful process took many, many, many, many weeks. Everything was done to scale so we could see the movement of the game and get a feel for the game's flow. I also had a little window cut out of cardboard that I moved over the paper to view the flow of the little screen."

As always, part of what makes a design succeed or fail is the interaction of the people on the team. In-house production teams seem to range from nightmares to dreams. Mike has seen both. "It really depends upon the group and the project and how the responsibilities are divided. I've done projects where the designer maintained more of a producer role, where he was just responsible for ensuring that everybody did their job. I've also had projects where the designer had more of a movie director's role of making sure that everybody is cooperating on the project and bolstering people up.

"Game design involves conceiving, planning, managing the product, and if you're working with a large group of people, ensuring that tasks get done. Of these, the only interesting part is the actual creation of the design, which involves a lot of trial and error and brainstorming and sketching out ideas and doing stuff like that. That's the most fulfilling part, but it's also usually the smallest part. The most frustrating part is having to compromise the original vision to fit it into a product that will actually ship."

Berlyn's work style tends to vary with where he is in the project. "I may go to bed at 9 or 10 o'clock, so, as a morning person, I could get into the office at three in the morning. I have two work modes: absolute silence and completely alone—which is good because I can do it early in the morning before anybody comes in—and I like to work with music, a lot of noise, distractions, feedback, and a lot of interaction with people."

When asked what advice he has for those who want to become designers, Mike semi-jokingly suggests, "Don't!!!" But if you persevere, he suggests getting as broad an education as you can. "If

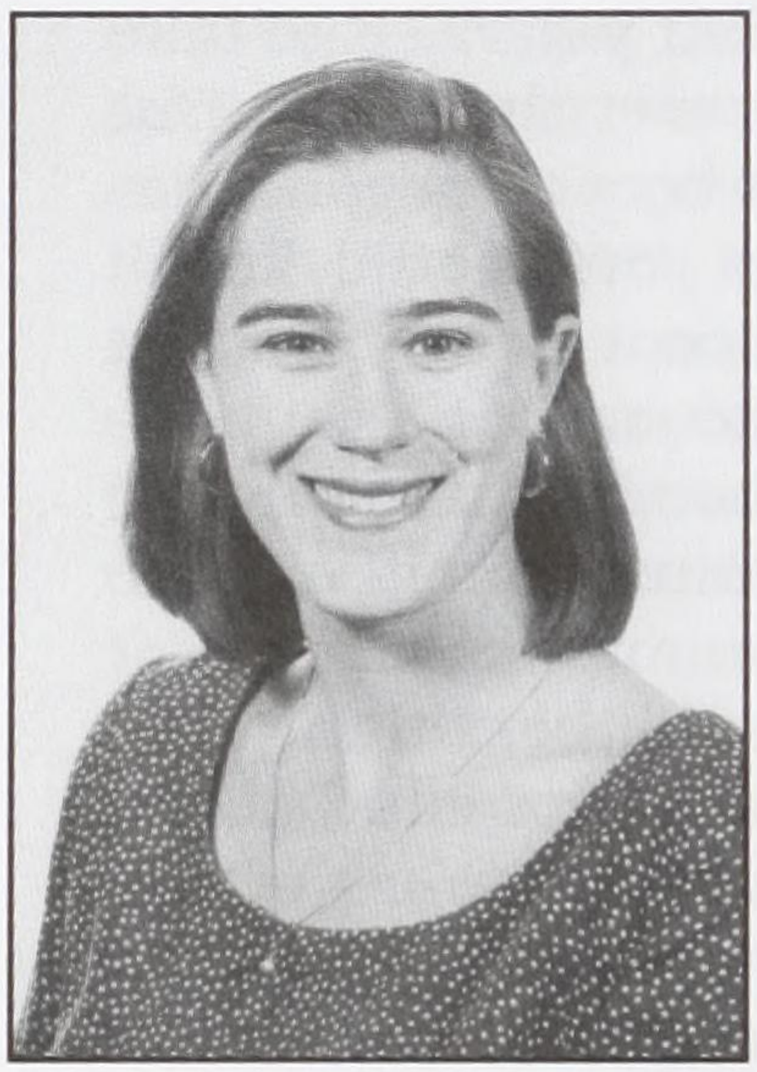
you want to be a doctor, there are courses that you need to take because they have been deemed necessary, over the years. But there's no course of study that will help you become a game designer. There are some fundamental things you should know: some computer programming, at least in concept; some basic art skills; and some basic music skills. Otherwise you can't communicate with other people about what you need, what you see, what you hear, and what you need to have done. You either have to be able to do all of that yourself or to communicate to other people what needs to be done, and I think the best thing that would prepare anyone for that is a good liberal arts education.

"Designing computer games or video games or games of any kind has little to do with computers," Berlyn says. "It has to do with understanding why people play games and what makes games fun. I mean, there are fundamental issues that knowing a lot about computers have nothing to do with. I've worked with brilliant programmers over my career who couldn't tell you why one game was more fun than another; they couldn't quantify it or analyze it. Strong analysis tools are very important, and a broad range of experience is important. You have to know what the computer can do—I program most of my games myself—and it really helps to understand what can and can't be done in terms of game designs, but the design process itself has nothing to do with computers."

Berlyn sees the future of games reflecting Hollywood's influence. "Interactive multimedia products will become more and more accessible as more and more designers turn their talents to that area. We'll see games in the den or living room entertainment area where game design plays a role that is as important as the actors, the directors, or content."

Julie Carlson

Julie Carlson turned her BA in East Asian Languages and Civilizations into a career in interactive entertainment. Since 1990, she has worked on a plethora of products for Koei. Julie translated and edited the manuals and game texts of *Romance of the Three Kingdoms II*, *Nobunaga's Ambition*, and *Uncharted Waters*. She assisted in localizing Japanese titles (*L'Empereur* and *Gemfire*) to the American market and, in addition to the



Julie Carlson

writing and editing, supported marketing. In her position as Koei's supervisor of research and development, she's handled not only the localization but also game designs for original scenarios. These titles include *Aerobiz*, *Romance of the Three Kingdoms III*, *Genghis Khan II*, and *Celtic Tales: Balor of the Evil Eye*.

Julie reached her current position through a nontraditional career path. "I started out at Koei as an interpreter/translator. I worked on our manuals—in Japan—translating them from Japanese to English, and while I was working on the manuals I had to play the games extensively. Soon you see what you like and don't like in games.

"I learned about game design in Japan. When I came back to the States, I started moving away from the manuals, toward designing the look and feel of our games. Since I read and write Japanese, I was originally going to take our manuals and do much more in terms of books and guidebooks, translating them from the Japanese products into American and kind of reintroducing the books to our American audience."

But before the documentation project could get going, Julie found herself moving into the game design end of the business. "Koei is a great company for allowing you to move into areas that you're interested in," she attests. "So I was able to move into design pretty easily. Just by zipping my skills in working with the games, and working with programmers."

Genghis Khan is a good example of the challenges and frustrations of converting a preexisting Japanese game for the vastly different US gaming market. There were several things Julie enjoyed about working on the project.

"Something that's particular about *Genghis Khan* compared to our other products is that we have the characters speak their lines. We don't have voices, recorded voices, but their mouths animate and their faces show emotion as they speak their lines. So, because my background is in writing, I think working on the text was fun. However, we did have a time limit on that and you

can't always get in all the technical details you want to. As for working on the flavor of the text, *Genghis Khan* is a strategy game. You can't really do much in terms of changing the graphics when you're reproducing a conversation. My job involved changing the text, changing the interface, and looking for graphics that we needed to work on. I was able to work on them a lot. Also, I actually created two very minor graphics for the game: the two ruler markers that appear next to a character's name, which show how the character is related to the ruler by birth or marriage."

But not all the changes went smoothly, and some problems simply couldn't be fixed. As Julie says, "The most frustrating thing is knowing that there are weird things going on with the interface and not being able to change them. I was reviewing the game again today and it was just hitting me in the face again—there are some weird loops! For example, when you want to go to war, you select if you want to lead the war first, then you select your units for the first troop that you're going to lead. Usually at that point I figure out that I don't have enough money to go to war anyway and I want to back out. But the back-out loop is very crazy and you end up going into steps that you had initially gone into during the command sequence anyway. That was an interface we weren't able to change and that's annoying. The majority of the people in the office are saying, 'You know, this is crazy. You've got to change this.' But you don't have the time to change it."

Julie sees game design turning more toward multimedia and increased interactivity. From Koei's perspective, "Our games are trying to reach out more toward the multimedia audience. I think we're going to see more interactive strategy games in the future. Games like *Genghis Khan* could really be excellent in the battlefield scenario. For example, you could improve the command where you visit your family. In the original game this was an extremely lewd command, where the player, if they're playing Genghis Khan, can call their wife and seduce her. You're able to sleep with her, and if everything works out right, in nine months you have a child.

"In the American version, we cut that and turned it into a sequence in which the character spends some time with their family, and we cut out all the scenes where there's a woman undressing and turned it into something really neutral. I think you

could turn it into something that goes back and reflects the families of the tribes in Mongolia, for example.

"For *Celtic Tales*, we've had great ideas on how to work rune spells, but given time limits we can't come up with the graphics and the program's not quick enough." Nonetheless, Julie hopes the move toward CD-ROM will help eliminate such frustrations. "As we head toward multimedia, I think that things would go much faster. Games should really speed up with the new systems."

At the time of this interview, *Celtic Tales* is Julie's current project, and she starts her day by checking the faxes from Japan, which contain revisions of the concepts she sent the night before. "I spend the first half of the day making all the corrections. The second half of the day is preparing new ideas, working on the game scenario, and generally improving the look and feel of the story. When I'm making corrections, it's generally a unification problem: The step-by-step interface that I've laid out for how a command should work doesn't, or the layout or graphic sizes I've provided aren't quite right."

Julie admits she is not an early riser. "I really work best in the late afternoon, after all the noise has gotten out in the morning, just a very low background hum of voices. And I tend to tune people out pretty well. I don't know if you've seen our offices before, but we don't have walls. We don't have cubicles either, so the noise travels, and I've just learned to tune out people. I guess that can be annoying with someone tapping at my desk trying to get my attention, but . . ."

Sometimes the tapping comes from other specialists on Julie's team. "With programmers I have to approach them in a different way; I can't always jump into a conversation with programmers as quickly as I can with writers. I haven't quite figured out why yet, but there is a skill in speaking with them when you're working and testing products together. What's frustrating, though, is that each person works on his own thing and has his own personal style. When the styles aren't the same we always end up having problems. I'd say some people work better in the mornings, some work better staying after hours, making it difficult to converse about multiple solutions. We generally schedule time when we're going to plan to talk about solutions to problems."

Julie advises designer wannabes to do what she has done, entering the game creation business through the back door and learning on the job.

"I would say, if you're really interested in working on games, find any way you can to get into a game company and then work your way up. If you can't get to the level you wanted, find an in and make yourself obvious.

"I am a true believer in action, in seeing people make themselves stand out. That's how it works at Koei—people make themselves stand out when they're moved to a new position, and I think that can happen in the game design world. It happened within our company, whether it was Japan or the United States; people shift positions or—and I've seen it happen in other companies, such as Sega—game testers advance to game design positions."

While Julie supports the idea of computer education classes, she advises that they be balanced with design classes as well. "Computer engineering alone isn't going to allow a person to slip into game design. It's good to have a well-rounded background in design so that you have a perspective on your games. If you're looking at a game just from a programmer's point of view, you're not going to be able to, I feel, be able to associate with the players who go in there and expect to be told of the game. Storytelling techniques, and graphical perspective techniques, would be really valuable for a designer."

Julie embraces the idea of game design as an art form, with a qualification. "It's really a collective form of art. It's not just one person's product nor even a group of five, because there are test groups and other people to bounce your ideas off of. You're going to have the guy who stays at home, the executive who only comes home on the weekends to play with their machines, you really have a wide audience. It's hard to target everyone in a game, but ideally you want to. So, it's kind of a community effort when you look at the design community."

Julie acknowledges that game design is still primarily a male-dominated world, but she feels that it is still accessible to women. "The majority of the people working in this industry seem to be men: You go to a conference and there seems to be a lot of men. Games though are for everyone: for people of all ages, ethnicity, and gender. People need to approach gaming with open minds, especially since the whole purpose of games is to bring people

together for fun. So, women shouldn't be afraid of approaching an industry that is typically male-oriented."

Phil Goro

Born in Hogansberg, NY, in 1968, Phil Goro's interest in computers dates back to age ten or eleven when he received a TRS-80 as a gift. "I started playing games," he recalls, "but couldn't find what I was really looking for: a good fantasy role-playing game. In high school, I learned programming and, through coincidence, bumped into a friend at Westwood Studios, a game development company in Las Vegas, and wound up going to work there. It wasn't really my intention to enter gaming—because everybody told me you can't make a living writing computer games."

Now, Phil has an extensive interactive entertainment pedigree. His career has evolved from programming—"I was the only coder on *Battle Tech* (Atari ST)"—to designing, including *Nightmare on Elm Street* (PC).

"In the past, my job for Westwood has been primarily as a programmer. I was the lead programmer on *Eye of the Beholder I* and *II* and *Lands of Lore*. I was also one of the two designers. Bill Crum is perhaps a little more responsible for the design than I am, but I tended to give him guidance because I had more experience in the industry than he did."

Phil is now working on *Lands of Lore II*, and he has done some minor work on *Kyrandia* (PC/Westwood) and *Hand of Fate* (PC/Virgin).

Phil is a firm proponent of game design as an art form. "There's no science of building a game; it's more of a feel sort of thing. The people who are best are those who can go through a game, play it, and determine balance issues well. For example, there have been a lot of games that I have played that just don't seem well balanced. There doesn't seem to be a set order in which you do things or any sort of learning curve for the player. I think an example of a good game might be *Civilization*, though the interfaces are a little overwhelming at first. You slowly build up and get more units—different units with more capabilities—as time progresses, so it gives you ideas and chances to keep up with the game. So I would say that game design is an art form. It's anyone's guess as to what the best approach is."

Phil adds, "My primary feeling is that we're all basically an entertainment industry, and that if something is entertaining you can do it in a game. I'm not so much into games for realism's sake, but more of 'just how do we have fun?' My philosophy in approaching the problem of realism is: 'Let's do it here, but be more.'"

Phil feels the visual element is very important in game design. "It takes a really concentrated eye to figure out what is really flashy and what also carries value. When you're designing a game, you effectively have to visualize yourself within the game world, and figure out all the elements: How is the art going to work with the sound, with spoken dialog, and that sort of thing.

"I like the concept of being able to layout and create a world. Specifically I like to go through a game design that has already been done or is in progress and critique it, see where areas fall down, where areas stand up, and see what we can do to make it better. My primary focus is then to improve on a design."

Conversely, Goro finds that starting a design from scratch causes him the most pain and frustration. "Basically I'd say my base weakness is dialog. I'm not very good at dialog, and I don't like writing spoken dialog. I can look at other people's dialog and tell you whether it's good or not, but I can't necessarily tell you how to fix it. That is my biggest headache."

Like most game designers, Phil has his own working style. "I basically work best under a very tight deadline, when there's all kinds of pressure on top of me to get things done. I tend to produce the most that way. I don't need to go off and find a place to be creative as much as I just need to get focused and attack what problems exist and deal with them."

He says working with other specialists can be "very rewarding and very challenging. I think that designers aren't always appreciated as much as they could be, and a lot of the mistakes are made on other ends. Basically, it is a set of checks and balances for art. A design will come through a straight process, with somebody determining at some point whether the concept is good or not before it gets to the game. By the implementation point, you're generally pretty much guaranteed to have good art. You're generally guaranteed that you have a play-testing department to help debug code. The designers debug their levels basically by playing through them and seeing what they forgot."

And if you want to get into game design? "If you want to be a programmer, then a degree in computer science or computer engineering is probably the right way to go. But if you want to be a good game designer and create good stories and plotlines, then an English or theater degree might be a better approach. Our in-house designers don't deal so much with technological issues as they do with a story: Is it a good plot, is the right level of conflict maintaining, are there the right number of and the right feel on the out-takes—or do we have too few or too many—and finally, are we getting any bang for our buck?

"Second, people should be sure that they are the type of person who is willing to deal with pain, willing to look within themselves and be very critical of their own work, because I can guarantee that someone else will be. Designers can save themselves a lot of trouble if they have been critical first and fixed many of the problems. Third, don't ever assume that you know everything about game design because someone will always come with something better."

Phil sees games moving more toward multiuser interactivity. "My thoughts are that games will slowly push toward being interactive and multiplayer games, which I find very exciting. I have done some work in the past for a multiuser bulletin board service, and I have worked on a couple of multiplayer games. I like a game where I am interacting with others, I think that's really best. I think that's the only place that you're going to get that. There's an automatic aspect of replay value in those games because you don't know what the person you're interacting with is going to do, if it's somebody new each time.

"I think that as we continue to develop as an industry, the focus is going to come more and more on either action games—games that just have a lot of action value and a lot of adrenaline, for example, *Doom*—or on role-playing games. Storylines are going to become more key and people are going to expect more. What we see today is going to become a lot more refined."

Phil's days are very full. "I get up in the morning, come into work, and the very first thing I do is I read all of the electronic mail that I've received in-house to find out if there are any pressing items. I will then sit down and look at my task list as to what comes next, completely ignore that, and start designing what I want to design.

"Well, that's not entirely true. I come in and I try to primarily focus on whatever is on my task list, whether it's designing another level for *Lands of Lore II*, whether it's trying to solve a technical issue because I used to be a programmer. I generally do take lunch but I don't make it out of the building, and depending on what mode we are currently in on a project—whether we're on start up or we're in crunch mode—I either go home eight to ten hours later or fourteen to eighteen hours later. When a project really gets into crunch mode, you find that you have to put in a lot of time to keep things going to get their product out aggressively. Anyone who wants to get into this industry needs to make sure that they're willing to dedicate that sort of time."

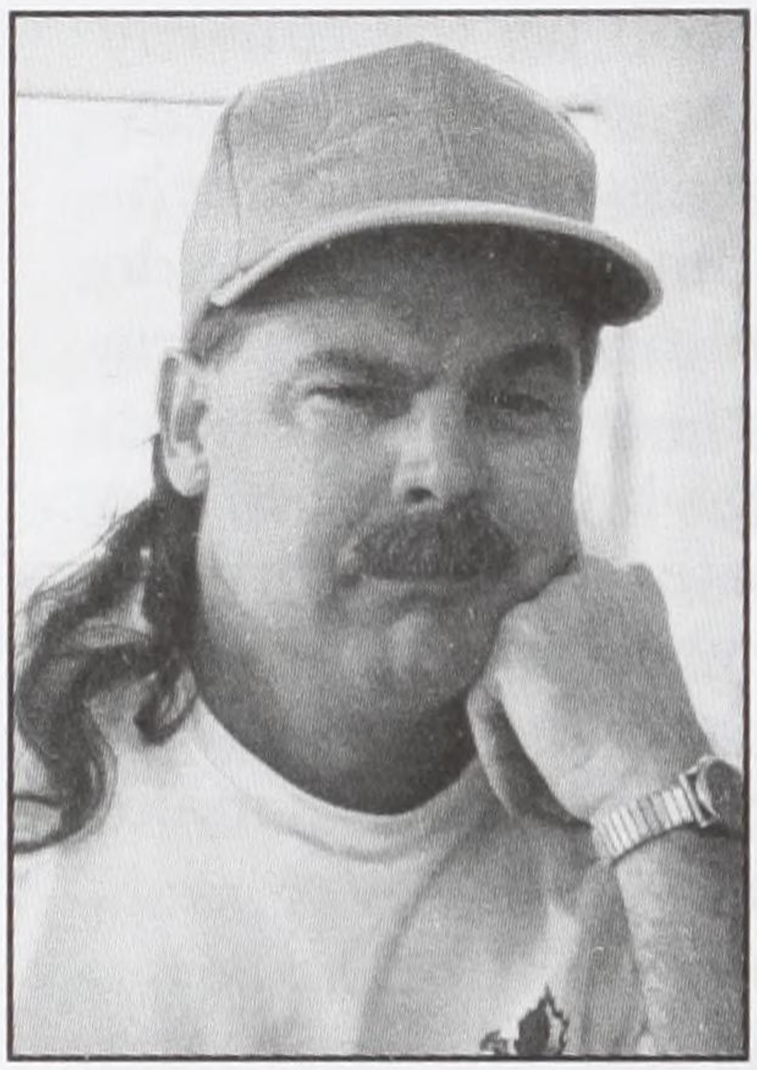
Rick "Coco" Gush

Rick Gush, known to his friends and coworkers as Coco, is an aging and marginally irresponsible Peter Pan who simply refuses to grow up, a determination that has served him well in a virtual Golden Book of Interesting Occupations. Armed with an education in horticultural pathology, he has also found employment as a rodeo cowboy, a uranium miner, a sensuality counselor, and a writer of musical comedies. "I like to do new and challenging things," says Coco. "I sold encyclopedias door-to-door in Mexico City for a few months, and I didn't speak Spanish when I started the job. Big fun!"

Rick's popular botanical treatise, *Gardening for the Criminally Negligent,* continues to sell well, but in 1992, he turned his full attentions toward creating games for Westwood Studios. "It's the new stage," Gush says. "It's the perfect medium for the modern playwright." Reputed to be the best speller in the Virgin organization (Westwood's parent company), Rick was a natural choice to become a computer games storyteller, and he has parlayed a fertile imagination and grammatical wizardry into his current position as one of Westwood's leading writer/designer/producers.

Rick's current résumé includes the PC versions of the *Fables & Fiends* series: *Kyrandia*; *Kyrandia 2: The Hand of Fate*; and *Kyrandia 3: Malcolm's Revenge*. He also wrote the text for *Dune 2*, which "had some of my favorite lines in it."

Rick clearly loves what he does, but he also takes it seriously. "It strikes me that there are a lot of people, even people who



Rick Gush

should know better, who think that game design is easy. It is extremely difficult, and I find it extremely annoying. Only in the rearview mirror is it any fun.

"I suppose that devolves to my feeling that I'm never as good a writer as I wish I was. I think: 'Oh, I'll make the ending work like this, and this is where I'll do this, and I'll have this neat feature here.' In the original concept phase, you can gloss over things like this with your expectations, but it's like golf: The instant you hit the ball there's nothing but reality to deal with, and that reality is always less than you envisioned.

"In my experience, the end result is always less than what you could conceive of. There are a number of reasons why, but it's just not that easy to sit down and write things. But at least ten times a week I'll have somebody either send me a design or make a phone date to talk about a design that they have. Without exception, I have yet to come across anybody who really has a concept of how difficult it is to design a really good game. People come up to me with their game designs—full game designs that are all essentially copies and/or minor bastardizations of an existing product. They say: 'I want to do it just like such-and-such game, except make everybody left-handed.' That's somewhat annoying to me because it implies that game design is easy and all we do is change a couple of things from games that are already out there, which is certainly not the case.

"That's probably the most important thing I have to say about my experience as a game designer: It's really hard, and the difficulty is compounded by the fact that you can't just write with the expectation that everything you produce will be wonderful. When I was doing musical comedies, I wrote on the 99-to-1 basis, which meant that I would write a 100 pages and throw 99 of them away, leaving one page worth of usable ideas. People think I'm exaggerating; they seem to believe: 'Oh gee, he thinks of several ideas and then he picks the best one.' I think the ratio is actually a little bit higher in the times that I've actually done a mathemati-

cal analysis of my writing, whether that be for game designs or for a musical thing."

Gush found his way into the game creation business in a most prosaic manner: the newspaper want ads. "Westwood Studios advertised for a storyteller. They recognized that the character development and dialog was not up to snuff. Prior to that, it had been essentially the programmers who were writing the dialog. They wanted somebody more professional to craft the dialog and make sure the story was told in the best possible way."

Gush also confesses, "I'm not a great playwright, but because I've written so much dialog my skill level for creating dialog was better than the programmers. And so I was just at the right time at the right place, and it has worked out fairly well so far."

Rick exults in the relative freedom a career in game design offers. "I have a dualism. I work very close to a nine-to-five job in the office, but it is almost all administrative responsibilities—either meetings with the programmers or the artists—although I do print up documents and do some design work. All the creative stuff I do at night, when I can put my feet up and smoke a cigarette and have a cocktail. I get into a groove where I can be real nice and creative, so I spend probably five to six hours at night writing."

Rick needs to enter into a different mode of thinking when he's writing dialog or storylines or developing characters. "I find it essentially impossible to do that here in the office. I have to be in my own little isolation cocoon to do the more free-form writing. When I'm in the office, I'm Mr. Computer, but at home I am a low-tech, stationery junkie. I have every writing instrument and every type of paper and folio known to mankind. Almost every weekend I go somewhere and buy nifty new paper, or a new kind of pencil, or whatever. I find a strange connection between my productivity and my excitement over the physical, graphic materials that I'm using."

Gush finds this tactile connection to his materials helpful when combating writer's block. "The lack of inspiration is a constant problem. To get myself in the mood, I change stationery in a way that gets me really excited. For example, if I got some brand new graph paper from Germany and a really cool Mont Blanc pen with purple ink in it, that excitement alone will make me want to sit down and start writing things. Once I've gotten past the bar-

rier of, 'Gee, I can't think of anything, and I'm just sort of scribbling on the page,' then I get my momentum up and go forward. But you can't repeat that same experience and still be inspired. It's the newness or novelty that's important to me, so I have a gazillion pens and different kinds of paper that are no longer in favor."

Rick has a lot of advice for people interested in a career in game design. "Number one, I would recommend gaining a general familiarity with the industry. That sounds like very basic advice, but a lot of people who weren't previously in the game industry are flooding in now, and I tell them, 'Go get the computer gaming magazines, start reading them, and get up to speed on who the players are.'

"Unfortunately, it's like so many other kinds of education. You are best off getting a broad scope of education, and I would think a general liberal arts background would be of greater value than one restricted to computer skills. I hate to say it, but you can always hire people with the computer skills, but you can't hire people who have the intellectual grasp to see the whole picture. If someone becomes a computer whiz but doesn't have a clue who Milton was or why *Paradise Lost* was an important piece of literature, then a part of the human experience that could be brought to bear effectively on game creation is missing.

"My biggest disadvantage when I entered the field, however, was the fact that I was essentially computer illiterate, so it took me about a year to get up to speed and know what people were talking about when they would say, 'Oh, the 32-bit library is ready to be installed.' I think having your feet in both of those puddles is really the ideal situation."

Rick praises his coworkers at Westwood Studios for their help in his making the transition. In an industry where large egos are the norm, finding a group of people who excel at working together can be like looking for an honest used car salesman. "In general, Westwood's a dream. I've had a lot of jobs, and interacted and known a lot of people and, without a doubt, these people are a remarkable breath of fresh air. There are essentially no big time politics going on, and basically these people are intellectually above the norm with the tendency to just be so simply straightforward. Their attitude is 'Okay, what's the problem, and how do we solve it?' It's like, 'Gee, we've got a job to do, so let's talk about what we could and should do, and let's do it.' That's

probably the single most enjoyable thing about my job. People think that the most rewarding thing is getting to put my name on a title, but on a day-to-day basis, what makes my job fun is the reasonableness of the people that I deal with."

The process behind the development of the *Fables & Fiends* series was fairly straightforward. "Brett [Sperry, president of Westwood], Phil Goro, Bill Crum, and I had a series of discussions, where we laid out some of our basic desires and talked about selecting some topics." This series of roundtable discussions didn't try to define the big picture, but they dealt with developing exciting ways for the player to enter the game and creating interesting characters to inhabit the simulated world.

However, this method of group creation was not working out. "After a few of those, it became apparent that good design isn't done by a committee. There's just too many disparate ideas, and somebody has to be in charge of the design. I finally said, 'Okay, I'm going to step up to the plate and be in charge of the design.' We stayed in that situation for six to eight months, with me carrying the title of producer and actively participating in design and implementation. *Kyrandia* probably took about thirteen or fourteen months total."

Gush also oversaw the character and story development, while Bill Crum handled the puzzle creation. "We had these two very distinct areas of design going forward. I'd say, 'Gee, this character is sent to a swamp and meets these people, from whom he gets this item. Let's make it so that if you kill a swamp creature, they turn against you, but if you're nice to the swamp chief, then they're nice to you.' Bill would say, 'Well, I want to have this monster chase you in this area, and there's a couple of levers that you pull to open a trap door through which you can escape.'

"All of the ideas that Bill and I came up with had to go through Phil because Phil was the programmer who was installing it. If Bill said, 'I need the pressure plate to be here, and if you step on it such and such will happen.' Phil would say, 'Now wait a minute, what happens if you step backward onto the pressure plate?' He would discuss with us the means of effecting the design that we had written. So on a daily basis Bill would be designing his little puzzles, I would be thinking about the story, and both of us would be in discussions with Phil."

Rick feels that games in the future will involve characters that are more sentient and will track information about previous relationships and other additional complexities. "Right now, most everything uses toggles: 'Gee, you were friendly; therefore you get a clue.' But, real-world relationships are considerably more complex than that, and I believe we will start using more verbs in an effort to craft greater programmatic diagrams and usable relationships."

David Johnson

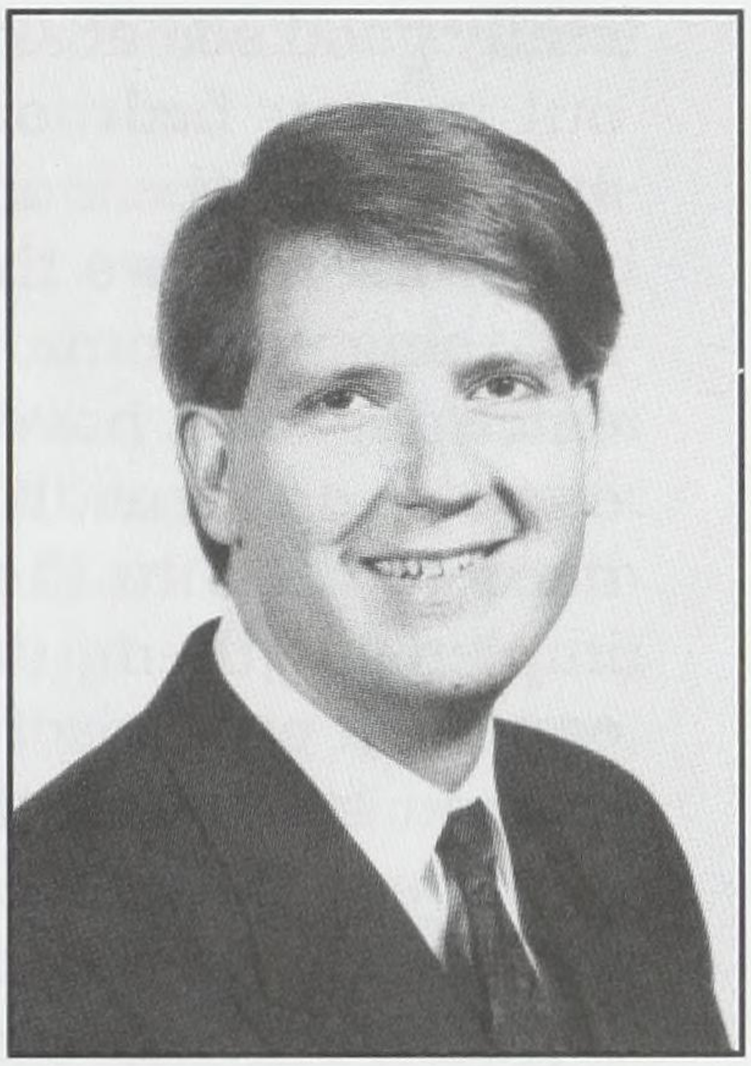
David Johnson became interested in game design when he was about twelve. "I used to live in Olympia, Washington, and the local state college, Evergreen, had computer systems available for anybody to come and use if they were interested. My friends and I went to Evergreen, we obtained user accounts, and we played games on their main frame, such as the original *Adventurer*, and some Evergreen-developed games.

"I then bought a TRS-80 color computer. When I was about thirteen or fourteen, I learned how to program in BASIC, and the first thing that I wanted to do was create a game. So, my friend and I got together. We drew some art work, mapped out the little world, and created a whole plan for this game. As time went on, it never actually developed, but it did spark my interest in game design. From there I bought other computers and learned more about programming, et cetera, et cetera, and here I am: designing games for a living."

Before joining Koei in 1989, Johnson worked for Under Contract, converting computer games to the Amiga, Macintosh, and Atari ST. When he joined Koei, his first project was converting and redesigning *Bandit King of Ancient China* from the PC to the Amiga. He also created the code for Koei's other Amiga conversions (*Genghis Khan* and *Nobunaga's Ambition*). Since May 1990, he has been the US development manager for the majority of Koei's releases. His current pride and joy, however, is as conceiver and codeveloper of *Celtic Tales: Balor of the Evil Eye* (PC).

David taught himself programming, but he did attend both the University of Washington and UCLA. While he graduated as a math major from UCLA, at UW he was a pre-engineering major with an interest in going into computer science. "My interests

were always in programming. I seemed to have a natural knack. When I first got my TRS-80, most programs were written in assembly language because there weren't C-compilers until later, so I wrote a disassembler to figure out how other people did things. I wrote that in BASIC, so I could look at their code and figure out basically how they were getting these characters and processing, or how could I modify the game to be better."



David Johnson

David reflects, "I think now, in retrospect, that my math major at UCLA was really important because when you're trying to simulate the world, you need to have a math foundation or similar foundation. There's two types of games out there right now: simulation and interactive fiction. Simulations require basically modeling the world, which requires math. Interactive fiction encourages people to go into writing or film studies."

However, David feels that initiating a career in the interactive entertainment industry requires determining "where your talents lie. If you're more of a logical person, perhaps programming would be a good area to go into. I really believe film would be a really good area to study because we have a lot to learn from the art form of film, including scripts and the ability to tell a story. Creative writing I think is also a good area, since text and character interaction is becoming more and more important. We're trying to provide the user with a real rich environment to explore; to do that you have to tell a story, and do it well. So I would say creative writing, film, and even theater are also good areas to look into.

"There's also the engineering role: the camera people, the artists, and again, the programmers. At Koei, we don't have a standardization, so we really have to work together, and the job role divisions can't so easily be broken up. You have to work closely together. For example, you have to work with the artist to know how big the graphics are going to be, make sure they're the right size, have the right look, and enough frames. It's a human group endeavor that everyone has to work together. You have to have

really good social skills also to create games. I wouldn't say hang out in your bedroom and learn how to program. Get out there and explore the world. Improvise with people."

Johnson views the most fulfilling part of game design as seeing the game come together. "What's really fulfilling is the ideas emerging. You have a synergy when you're working with someone else on ideas. It's extremely exciting at that point. When you need to get into the details and explain to other people how to implement them, it gets a little bit less exciting. But once you can see it on the screen, that's when it really comes together and you get a sense of the world you've created, and it's really exciting to see that in place."

Working for a Japanese company can sometimes be frustrating for US designers, both for practical and for cultural reasons. "With *Celtic Tales*, we have an international team, and what's really difficult is that we're in two separate physical locations. It wasn't planned that way, and it wasn't supposed to be that way, but that's what happened. So keeping communication open between the implementation team in Japan and the two primary designers here in the States is sometimes a real challenge. Infighting can happen and arguments are sometimes draining. It's very personal, and it's a product that belongs to each of us. We really have some strong ideas on what it should be, but there can be only one product that comes out of it."

However, for the most part, David finds the collaborative process very helpful and stimulating. "With *Celtic Tales* it's been really exciting to get the first draft of the music from our composer. Then we had to put it onto the sound board. We basically got minidata from the composer that was made for one sound board, and we needed to port it over to the Sound Blaster, and some of the other sound cards out there. It has been really fulfilling to hear that music and see it evolve into a fantastic sounding sound track on those cards.

"The same goes for the artwork. It's been really exciting and wonderful to see the artwork grow and expand and see the animation. The animation in *Celtic Tales* is really entertaining. It's really neat to be able to see these people walking across the screen and doing things in their day-to-day lives. It's very fulfilling.

"Marketing is the same way. We got an amazing artist for the packaging, and I think that the package is going to look really

good. We're also discussing the logo creation. So those are some of the joys of working with these other specialists."

In the end, David finds the time pressures the most frustrating part of the job. "We have to get a product out the door, and as designers, we want to make the best product possible. Unfortunately, sometimes we have to say, 'Okay, this is it, we can't do any better,' even though we want to. The artwork and music need to be done by a certain time. Marketing wants to get the product and know when the product is going to be done. It's difficult getting it all together and pleasing everyone, finding a balance between what is really the best possible and what people want."

David feels that, in the future of game design, simulation and interactive fiction are going to merge together—that is, they will once more powerful machines can solve the artificial intelligence problems that currently exist.

"Right now interactive fiction is basically limited to a decision tree and its pathway to a solution. Simulations are a little more open-ended—you have a world you can create or explore. When you merge those two, then, basically the computer is creating the story on the fly, and I think that's when it's going to be really exciting. When we have that, then it's going to be an amazing time."

David doesn't hedge when it comes to supporting game design as an art form. "Game design is certainly an art form. It's such a mix of different art forms. With interactive fiction, you're attempting to convey a story to the user, or in a real-world simulation you're trying to convey this information to the user. How to do that in the most effective way is a very creative endeavor."

Johnson's typical work day is extremely busy. "We do a lot of writing and talking about the idea for the design. Plus, there's the programming. I've been programming a map editor. In *Celtic Tales*, you have a world with various provinces, and I'm creating a map editor to edit the world, while other designers create the world that the player will play in.

"Mark Hughes, Director of Marketing, comes over and we talk quite a bit about game stuff. We go over the logo and decide on the type style and colors. We talk about the way the artwork looks and whether or not it conveys the appropriate images for the game. Because each little item is supposed to be a symbol of something, it must have the right look and feel to the rest of the

world, and convey enough information to the player.

"Each little unit can represent different levels of growth. For example, you can have a small tree or a big tree, depending on how long it's been in its location. When we were going over the music, we sat down with the sound designer, listened to the music for the Sound Blaster, and determined that we wanted to make the sound more dynamic. Then he changes that, and we would sit down the next day, listen to the changes, and see if that's what we're looking for. It's a very interactive process, and requires a lot of social communications. Game design is, as I said before, a group endeavor. There are programmers, sound people, artists, and designers, and you try to bring them all together."

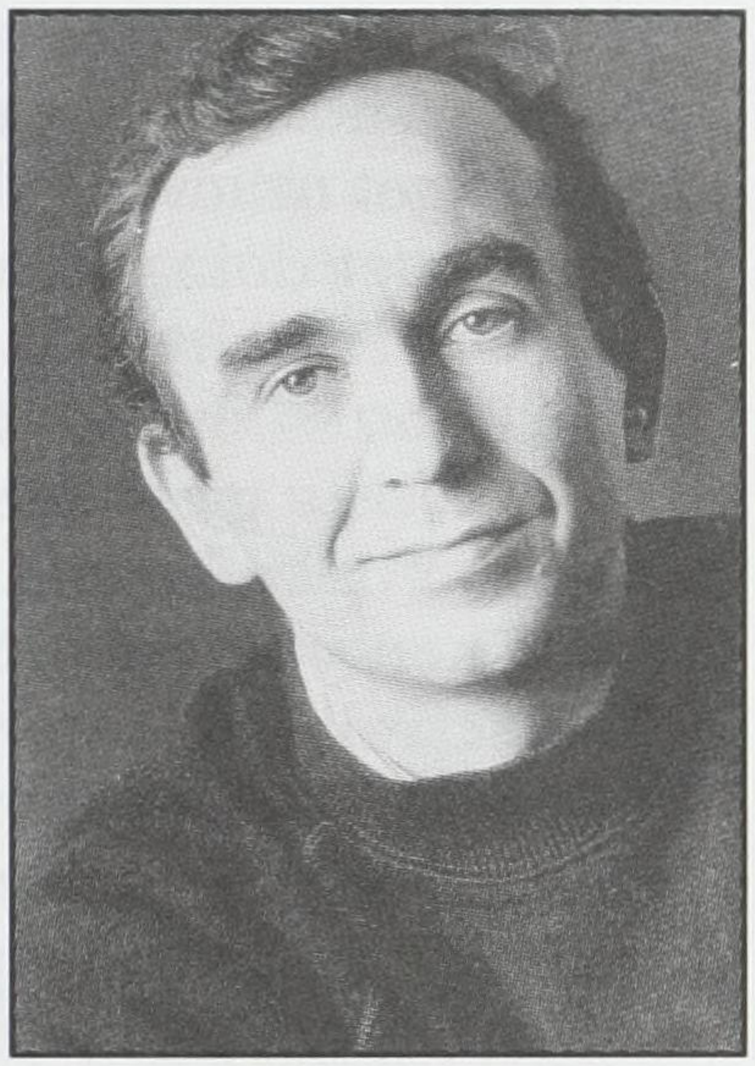
Peter Molyneux

Peter Molyneux loves to build, and for almost ten years he has built his own independent software-development company, which specializes in construction-driven entertainment software. Peter is the cofounder of Bullfrog Productions, along with Les Edgar, and he has been both a designer and a programmer on *Populous I* and *II*, *Powermonger*, *Syndicate*, and *Theme Park*.

All of these games involve the user in the process of building something, whether it's a world (*Populous I* and *II*), an empire (*Powermonger*), a corporation (*Syndicate*), or an amusement park (*Theme Park*). In fact, for Bullfrog's first major release, the god-simulation *Populous*, Molyneux and company physically constructed the early terrain models using Lego blocks ("It worked fine until we had to add water," he reports).

Molyneux has always been interested in computers. "I always had a computer. When the very first Pong machines came out, I set myself to get one. I've always been interested in board games as well, and it was just a natural progression for me to get into computer games."

However, computers have been more than a source of personal entertainment for Peter. "The thing that forced me into the computer game world is that I am terribly dyslexic. About the only thing that I can possibly do is write computer games because the computer tells me if I make a mistake. So it was a natural progression and profession for me. Thank God it's here. If it wasn't, I'm sure I'd be a bum on the street."



Peter Molyneux

Far from being a bum, this industry giant started his career with a degree in computer science and headed into the business world. "I started programming databases. After a boring period there, I started up Bullfrog Productions in 1987. We initially started production on a database package called Inquisition on the Amiga."

When the Amiga crashed and burned as a business machine, Commodore revived it as a game platform. "It was a case of sinking or swimming; so, we switched over from writing databases to writing games. The first game I wrote was a game called *Populous*, which was hugely, hugely, hugely successful, and the rest progressed from there."

Despite its ultimate success, selling *Populous* was not easy. The idea of a "god" game was a completely revolutionary concept in 1987 and thus commercially unproven. Even two years later, when the game finally made the trip across the Atlantic, it was still a tough sell. Maxis' *SimCity* was rolling off the shelves, but it was more of an urban-planning simulation, with the user cast as the city's designer-leader. *Populous*, meanwhile, makes no bones about the player's role in the scheme of things: He or she is a deity with the denizens of the new world constantly seeking to expand their "spiritual" dominion.

Nonetheless, when presented with the opportunity to obtain the US rights to the game, Electronic Arts agreed to take a chance and signed a one-product deal with Bullfrog, which has led to a long and profitable relationship for both companies. *Populous* was released to unanimous acclaim and almost two dozen awards in the United States.

Electronic Arts has stuck with Bullfrog, and in June 1994 they brought out the first title in a planned "Designer Series." *Theme Park* combines accurate simulation technology with gaming elements. It is also the current focus of Molyneux' attention.

"In *Theme Park*, you build, design, and run your own theme park. You start off with a big open field. There's nothing there. There's a bus stop outside the gate. You have to put down rides

like haunted houses, boat rides, race car rides, and all the other rides. Then, you open your park up and run it." And where does Peter get innovative ideas like this? "I usually get my game ideas from just doing normal things," he confesses. "I got the idea for *Theme Park* by just going to one, and I thought, 'This is a great park, but I think I could design a better one.' Then I thought it would be a great idea if everybody could design their own theme park. That's how the idea starts, and that's the exciting bit.

"Then you start thinking: I could put roller coasters in, and those roller coasters break down, and you could see the cars shoot off the side of the track! And make them as high and as fast as the highest and fastest roller coaster in the world! That's the really exciting part; the process of creating the idea. And then the most exciting part is just seeing it develop, from the first time you see those very first few pixels on the screen to the time when the game is actually finished. I mean, it's all pretty exciting, except the bit I'm doing now when you're fixing the glucks [bugs]."

"The most tedious thing is that it takes a long, long time," Molyneux good-naturedly complains. "Back in the early eighties, I remember reading a magazine about this one famous programmer, and in a week he produced twenty games! Those days are gone. They're gone forever. It now takes about four months to two years to write a really, really good game. I mean, from the very first conception to the time when you can see it on the shelves at your local computer store, it takes almost two years. And that's really frustrating because in that two years you've got to stop your mind from wandering off and creating another game, and another game, and another game, because otherwise nothing ever gets finished. So you really have to concentrate, and you have to make sure that it's finished, and it just takes a huge amount of time. The most frustrating part is right at the end. Three months ago if I were on the phone with you, I'd say, 'Yup, *Theme Park* is finished. It will be finished by the end of today,' and three months later I'm still saying exactly the same thing."

As with all designers, Peter's work is highly collaborative as well as solitary. "I really like working with people and there's now forty people at Bullfrog, and we all work as a team. But sometimes you just have to go away, turn the music up loud, and hammer into the machine. It really varies with what I'm doing. If I'm doing something really complex, then I much prefer to work

alone, with the music going. For most of *Theme Park*, I've either worked with one other person, or I've worked with a team of people. I'd have to say, unless it's something really complex, working with people is much better because you can bounce ideas off them, and if you get really stuck, you can always turn around and say, 'Hey, do you mind having a look at this with me?'

"I think the best thing is that there's a cross-fertilization of ideas. For example, you put something into this cooking pot of people, and it gets bounced around a bit and comes out as an even better idea. But that can also be the worst thing. In your mind, as a designer, you've got to have a really, really focused thought of what the game is going to be like when it's finished. Sometimes it's hard to say to people, 'Yeah, I know that this is not too good at the moment, but when all these other features are in, it's going to be great.' When you're working with other people that is really tough to do because they'll swear that you're making a mistake, and you'll swear that you're not. That's where arguments can come in.

"Thankfully it doesn't happen too often because you try to be as vocal and patient as you possibly can when you talk to people, so that they get the same idea. Sometimes you do get this clash of ideas where other people can't see what you see, or you can't portray to them what you're seeing, and that can be very frustrating."

Peter has seen game design evolve from the mid-eighties, and he sees it only getting more complex in the future. "Historically, if you look at computer games, they started really, really simply. I mean, the design for *Space Invaders* could have been written on the back of an envelope and probably was," he chuckles. "I don't know. But game designs now are hugely long; I've seen game specs that are over two hundred pages, which is equivalent to a film script. So game design is getting more and more complicated because the hardware that we're working on is more complicated, and the public demands more and more realism. I mean it's no good anymore to just have a shoot 'em up. You've got to have a shoot 'em up within a real world.

"As time goes on, people are going to demand more and more from game design. I'm working on a design at the moment called *Magic Carpet*, where you fly over a landscape on a magic carpet and cast spells. Now, within that game design, everything is as

real as possible. For example, you look up in the sky and you can see the clouds move over, and as the clouds move over the sun goes in and there no longer are shadows cast on the ground. If you fly down to the coast and look at the water, you'll see it lapping on the shoreline. If you fly down to the coast a few hours later, you'll notice the high tide mark has moved. And the realism has got to be built into the game design. Realism is fine. Realism is great, but you've got to balance that realism with a really good solid game design. In other words, making a game playable, or making it fun to play, gets tougher and tougher to do. It's no longer possible to say, right, here's my game spec of fifty pages, or twenty pages, or ten pages. You've got to play the game and live with the game and design the game as it goes on because it's a real world that you're working with, or as close to a real world as is possible to make in a game.

"Game design is just getting very tough. I mean, Hollywood finds it really hard to produce a good script. How many good films have you seen? You could probably count them on two hands. And now computer games are getting like film scripts, and the trouble is, the same is going to apply. How many good games, how many good scripts, are you going to see or play? It's going to get very tough."

In light of the growing toughness of game design, Molyneux proposes a more straightforward approach to entering the field of game design. "Almost all of the people that we've taken on have no prior experience because I believe that the most important aspect in a computer game career is enthusiasm. You can have degrees and PhDs up to your eyeballs, and you could have written twenty products before, but if you haven't got the enthusiasm, you haven't got anything. So that's where you should start from.

"Writers should just say, 'Do I really want to write a game? Is there a game inside me that I want to write,' or, 'Is there a game that I want to craft, to make as good as possible?' If you're an artist, draw sketches, and build up a portfolio of work, and send it around to anybody you can, and say, 'Hey look, these are some ideas I've had, or these are some visual ideas I've had.' That's a real easy step for artists. For musicians, the same applies. Just compose music, get a game that's already out there and recompose the music for that and just show how creative you are. If you're a programmer, it's much tougher because you're going to have to do more work to conceive a demo. But if you can, it's really good to produce a demo because if you come through the door or phone someone and say, 'Do you

mind if I send you this demo?' it just speaks volumes for you, if you can show something on the screen.

"If you're a game designer, then conceive a game, but don't try to conceive the best game ever. That's always the big problem: people trying to take on too much to start. Begin with a really easy idea, jot that down on a piece of paper. The key to it is—what really makes a difference, especially with other good game designs—is to just talk someone through the first two minutes of what the player sees and does. If you can do that, and I can get excited by the first two minutes of the game, that speaks much better than a two-hundred-page spec."

Peter is firm in his belief that game design is an art form. "A lot of people would argue that an awful lot of things are, from gymnastics to digging the road up. In my mind, anything that is created from nothing and influences and changes people's lives has to be pretty close to an art form. That's what computer games do: they change people's lives. I'm not talking about *Street Fighter*, or the obvious games. I'm talking more about the games that are much more involved, and have actual environments. That's much more clever and much closer to an art form than a lot of things."

Molyneux' work day varies from project to project, but he does try to establish a routine. "Normally, I try to work when everyone else works. So I get in at 10 o'clock in the morning and finish about 6 o'clock. And then, because I'm a rather sad person, I'll go home and continue to work after dinner, and I'll work another three to four hours, until about midnight. If it's really going well, or I'm trying to get an idea finished, then I'll work until two or three, if I'm really pushing myself. So, it's about ten hours of work a day. Maybe more."

Molyneux does have an admonishment for people in the industry, though. "Everybody in the industry says that game design is getting more and more stale, but it's not! You just have to think more creatively. You have to think about a broad audience. If you're trying to think of a game design, never, ever look at other games. Never, ever pick up a game and say, 'I could do that game better,' because that's not being creative, that's not game design. That's just improvement. Just be creative. Just use your imagination."

Scott Orr

Scott Orr joined Electronic Arts in 1990 as a designer for EA's first sports games for the Sega Genesis. In 1991, he moved into the role of a producer and is currently the vice president responsible for Electronic Arts' High Score production team. While he is most widely recognized as the designer/producer of *NHL Hockey* (1991-94) and *John Madden Football* (1990-94), he's also worked on *MLBPA Baseball* (1994), *Mario Andretti Racing* (1994), and *Bill Walsh College Football* (1993-94) for the Genesis.

Scott was hardly an industry novice when he joined EA, however. He started his own company, Gamestar, in 1982. "I wanted to be in the business, and the industry was sort of just getting off the ground. I was a real fan of television sports games, and I kept waiting and waiting for the Mattel sports games to come out in a form that included one-player versus the computer. All their sports games were two-player games only. When the Commodore 64 and the Atari 400/800 came out, I saw an opportunity to do sports games for something other than video game machines, and a business I could get into. My partner and I started the company. He was the programmer, and I handled the game design, marketing, and those kinds of things.

"It was really a business opportunity. My background was marketing, and I had, like a lot of people at the time, an MBA. I really thought I was going to go into some sort of marketing job. When I got out of school and returned to Santa Barbara, I hooked up with a guy who planned to start a chain of computer stores, back before there were a lot of computer stores around, so it was an interesting concept. I actually went to work as his marketing manager for a few years and in the process I met programmers."

As anyone can see from his résumé, Scott loves sports. "As a kid I played all the major sports. In college I played intramural softball and basketball, and I liked football. I've always been a lover of sports, both as a participant and a spectator. I was able to bring an understanding of sports to the design, and I was a gamer. But what really convinced me that I wanted to design and produce sports games was when the Intellivision came out, and they had the baseball and the football games."

Orr sees game design as a creative process. He candidly admits that working with the other specialists is "sometimes

fun, sometimes not, but it's a very dynamic process.

"When you work with people who share the vision, there's a synergy that's created. But in some ways, it's compromised because the synergy you get from a team of people who share the same vision can be different from a team of people who are just doing their job. So, you might end up with a competent product or maybe even a product that's good in its own right."

Scott Orr

Scott was fired up to produce the kinds of games he wanted to play himself, and to some extent he's had to learn to be satisfied with less than his imagination can conjure. "The most painful thing is realizing that your vision is not feasible on a particular platform, and so, by necessity, you have to accept a compromise or design changes to reflect the capabilities of that particular hardware platform. Over the years you sort of learn what the hardware is capable of and then set your design expectations accordingly. When you first get into the business you think, 'Gee, anything is possible,' but when you've been around awhile and gone through as many projects as I have, you have to accept that your vision may be more than the hardware can handle, and you really need to be happy if the essence is captured."

A strong vision and the ability to compromise are two important characteristics for anyone in the industry, Scott feels, especially when it comes to working with other people. "Back in 1990, I codesigned *John Madden Football* (Genesis) with Richard Hillerman. We had a lot of similar ideas, but different approaches: I wanted to create a very easy to play game that was very intuitive in its feel and really emphasized action; Richard, having worked with John Madden on the PC version, was more into having a very realistic and strategically strong game. We ended up compromising. In the process, we had several breakthroughs: passing windows, unique field perspectives, and graphic approaches.

"We started the process saying, 'Here's the new platform, here's what we can do given the technology, here's what the competi-

tion is like, we've got to beat them.' In sitting down and doing the initial design, our first step was laying out the features. Plus, we had to meet people's expectations: Everybody understands the sport and the rules you have to abide by. For example, football must be eleven-on-eleven; you can't knock off a few players here or there because the hardware can't handle that many sprites. You have to figure out a way to work around that.

"Basically, the first step was to lay out the features, and when we figured out the scope of the project, I started working on the specifics. I wanted to capture the essence of some of the other football games that I had done when I had *Gamestar* and when I was at Activision, but I also wanted to have a game that clearly beat the competition in terms of the look and feel. And that comes out during the development of the product.

"You can lay out features and the interface and give a general sense of the art direction, but you really want the specialists to do their things and bring their expertise to the project. Inevitably, the initial layout design changes significantly during the course of development. Sometimes you, or other team members, come up with better ideas, or you run into technical problems. There is a certain flexibility that you just have to accept. For young programmers that's difficult. I've seen a lot of guys come into the business, and they have great ideas. But they want to control the process too much or they're unwilling to compromise, and that makes them very difficult to work with. In a business where so much is team oriented, the designer has to accept his role in the project, and not think of himself as the Lone Ranger.

"In the old days, you had one designer and one programmer and everybody did a little of everything. You had a lot more control, but now there's typically two or three producers, two or three programmers, a couple graphic artists, and a sound musician. The designer is more of a director and idea guy who really doesn't control the project like the old days. People coming into the business need to understand that it's more like making a movie, and there are very defined and understood roles. If they need to control the project, or make all the tough decisions, then perhaps they should look into becoming a producer rather than a designer. And certainly the best producers in the business are those who are good designers as well."

Scott's advice to anyone interested in game design is to know the business, inside and out. "First, I would ask if he or she was a gamer. If the answer's no, I'd suggest they might want to do something else for a living. Otherwise I'd ask what kind of games they like the best. If it's sports games, for example, I'd find out what their favorite games were and ask them to think about what makes those games work, both commercially and for them as a gamer. Understanding the competition and why a product works is a major step.

"It's real easy for us in the industry to look at the competition and say, 'Well, that's okay,' or, 'That sucks.' When my producers look at a competitive product, I'm more interested now as to what they do—rather than what they don't—like. We're an industry where products are, almost by definition, limited because of the technology or money constraints, so it's real easy to pick something apart. But if someone has the ability to look at something and know why it works, why it's successful, and why people like it, then they have the makings of being a good designer.

"Beyond that, I suggest that they get their hands on every book about video games that they can. Spend a lot of time with the monthly magazines, and do their homework on the different games and systems. Basically, they should approach this as a profession rather than a hobby. And let's say they come up with a great design. Rather than putting that on paper and sending it off to all the publishers, time would be better spent going to local computer club meetings, where they can meet programmers, graphic artists, and other people they could team up with—and not only put a design together but potentially put a project together, because that's much more interesting to a prospective publisher than a five-page overview of the design."

Scott is not someone who does all his best work from 9 to 5 in an office. "When I was exclusively doing design work, I sat down every day and tried to get work done. But it wasn't like I got up at 8, worked for four hours, did other things, and then came back to it at night. It wasn't quite that rigid. Typically I had deadlines, so I couldn't spend three months or six months working on a design. We would have a month or six weeks, so having a certain process was necessary to meet those deadlines.

"Some of my best ideas would happen when I would go up to San Francisco and walk along the bay. It was really walking and

thinking about what the objective was that really led to the most creative ideas. An office environment is not really conducive to that. If you already have notes and you know where you're going, I'm sure you can sit in your office and think about the design document, but the document is really the easiest part of it. It's coming up with the ideas and putting them together in a logical fashion that is a challenge for any designer."

Orr's hobby is magic, and he finds that this has informed his design work. "Ever since I've been a kid, I have been into magic cards, coins, and close-up magic. That's a hobby of mine that I use to escape. It gets me off thinking about the business, and I think about both the mystery and the intrigue of how tricks work and how they're put together. In a sense, a relationship exists between the process of magic and the process of game design: Things have to be put together in a fashion which ultimately leads to either an entertaining or mysterious effect, or has an emotional effect on the viewer. The use of magic, and the whole process of understanding how card tricks work, was very helpful to me early on in understanding how a computer game would work." Scott believes that the designer will take on a greater role in future game development. "I think the days of the producer, and the designer, and the director, and the programmer are gone. Also gone are the days of the designers who typically did a very detailed fifty-page document that was very specific about the look and feel of a game, but then weren't all that much involved during the development. Those days, I think, are over.

"The demands, complexity, time, and money pressures of projects, and the stakes in terms of generating a hit, require designers to be much more involved throughout a project than in the past. That new role really requires more of a director feel and approach than a screenwriter approach, which is what the old days were like. The time commitment is something designers really need to consider and be prepared to deal with, because it's increasingly difficult for the traditional game designer to do a document and then walk away from the project.

"Now, as we move into multimedia and interactive movies, there will obviously be roles for traditional writers because of the story content. As the platforms become more sophisticated and the characters more fleshed out, there will be a role for those types of specialists. What they'll bring to our business are the kinds of

things they bring to movies or television. But the designer really has an opportunity now to emerge, the same way a director emerges in the movie business. It's not easy because video games and computer software have traditionally been a producer-driven business. It will be interesting to me to see who emerges over the next five years as a focal point of the media. Right now it's producers and programmers, but with any luck it will be the designers, since without the designers you don't have a product."

Jeff Peters

Jeff Peters became interested in video games back in the eighties when he was in high school. Currently in his late twenties, he remembers those days fondly. "I just started going to arcades a lot, and I started getting involved with video game tournaments that were being set-up. It sort of peaked my interest, and getting involved with the industry sort of popped into my head. I decided, 'Hey, I enjoy this, so why not make some money out of it, or at least a living.'

"When I was in high school I started buying coin-op equipment. I operated it, and by the time I got of high school, I was pretty much running an arcade. I then sold off my interest in that to start an entity called the US National Video Game Team with Steve Harris.

"We were in that for a couple of years. The USNVGT did a lot of domestic promoting and marketing and traveled to other countries for magazines. I eventually moved into publishing, did a coin-op newsletter, and eventually started what's now known as *Electronic Gaming Monthly*. After EGM started becoming successful, Steve and I sort of split up.

"I stayed in the video game business doing consultations and designs. I had a graphic arts company, and I worked in both the coin-op and the consumer industry. I went to work for SNK when they were finishing their NES titles, and we started working on the *NeoGeo*. From there, I went back to graphic arts and came to Salt Lake City and Sculptured Software. It was a pretty bizarre trail, but I guess I've worn all types of different hats."

Jeff's varied career is reflected in the games that he's had a hand in. In the coin-op industry, he worked on a number of SNK games, including *Beast Busters*, *Cool Pool*, *Nightball Shootout*,

and *Baseball Stars*. He's also worked on *Chris Stollis Baseball Stars* (NES) as well as the translation of Williams' *Mortal Kombat I* and *II* (SNES). One of his more off-beat products is an edutainment game that is being mass marketed by ElectroBrain called *Rex Roland, Experimental Surgeon*. "It's a very challenging product that basically teaches kids not to smoke. People can laugh, but I'll still claim the product."

Jeff believes that since "we're in the entertainment industry, it's our job to create something that entertains others. Just as filmmakers create a film, designers need to create an entity that takes people away from reality for awhile. There's a pride that the fruits of your labors have now provided entertainment for someone else. It is really a cool feeling, especially after spending so long with one given project."

Peters' single most dreaded foe is time. "If we don't have a contract with a given publisher, then we have X amount of time to execute a project. Sometimes—in fact almost all of the time—it's never long enough. Nine times out of ten a project is understaffed, and you never quite achieve the vision you started off with. That's probably the most frustrating thing.

"Since I came to work here at Sculptured, I've been fighting about placing the appropriate people on a project and how to achieve the player's expectations. The days of a six-month project are completely gone. Unfortunately, a lot of these people have been around for so long they haven't accepted that fact yet, especially the expectations of what constitutes a good game. Those expectations continue to rise and, being a development head, you have to keep meeting them, which unfortunately ends up requiring more time on our end."

Jeff finds assembling a good group is crucial. "I personally like to assemble a group of people who work very well together and who have compatible personalities and a group synergy. If that's the case, you can get a group fully motivated, and they'll maintain their interest. Which means you end up with good results.

"When good sound guys work with good artists, good programmers, and the other talented people who fit in between all those steps, you generally end up with something pretty cool. Unfortunately, that's not always the case. Often you get personality conflicts and that's hard, especially when you assume these people are married to each other for at least a year. To me

it's so very important to make sure that all these people can get along and that they're not just employees working together. Hopefully, these same people are going out and doing after-hours things together, showing that their interests are similar or on the same wavelength."

Jeff notes, though, that a lot of the designer's trials involve hardware limitations. "Doing something special with sound or graphics, or anything else for that matter, is challenging on the Super Nintendo or Genesis, both of which have incredible limitations. The PC doesn't have so many limitations."

When asked about the so-called black box systems, Peters hedged. "There is really an identity problem with 3DO. What is the consumer buying when they pick that thing up? You talk to [industry reps] and, to them, it's a multimedia machine that can engage in modem activities and play movies. The Philips CD-i system isn't that bad, although the access time is rather annoying. But Philips seems to be content with its little niche market which will probably always be one to two percent. Atari is at a very volatile point. Although actually the Jag is going to be an easier system to develop on than say, the Saturn. The Sega 32X is nice, but how many people who own a Genesis are going to put more money into it? Another hundred and fifty bucks to play better Genesis games, with the Saturn only six to nine months behind it, just doesn't seem like a wise thing to do."

Like most designers, he has two work modes. "During the day, I work around everyone else. That way we're getting the synergy flowing, making sure things are on the right course, getting new ideas that come up, that sort of thing. There's a lot of motivation gained from working with people who are still jazzed about what they're working on. My other mode is generally after people start filtering out, when it quiets down. That's when I personally get more productive, after all the hectic things during the day are through, when you can stop interfacing with people and get things done for the next morning."

Jeff has a few words of caution for anyone just entering the field. "First, gain a technical understanding of what it is you're working on, since without that you'll waste a lot of time. Second, be careful who you talk with and what you talk about. There are too many copycats in this industry, and dropping a simple comment to the wrong person means five companies are now work-

ing on your idea. That's true for a lot of industries, but with this one being so small, it really is amazing how quickly everyone is working on the same thing. When you have extensive deadlines, a year is a long time to keep quiet about something, and for someone else to catch up to you.

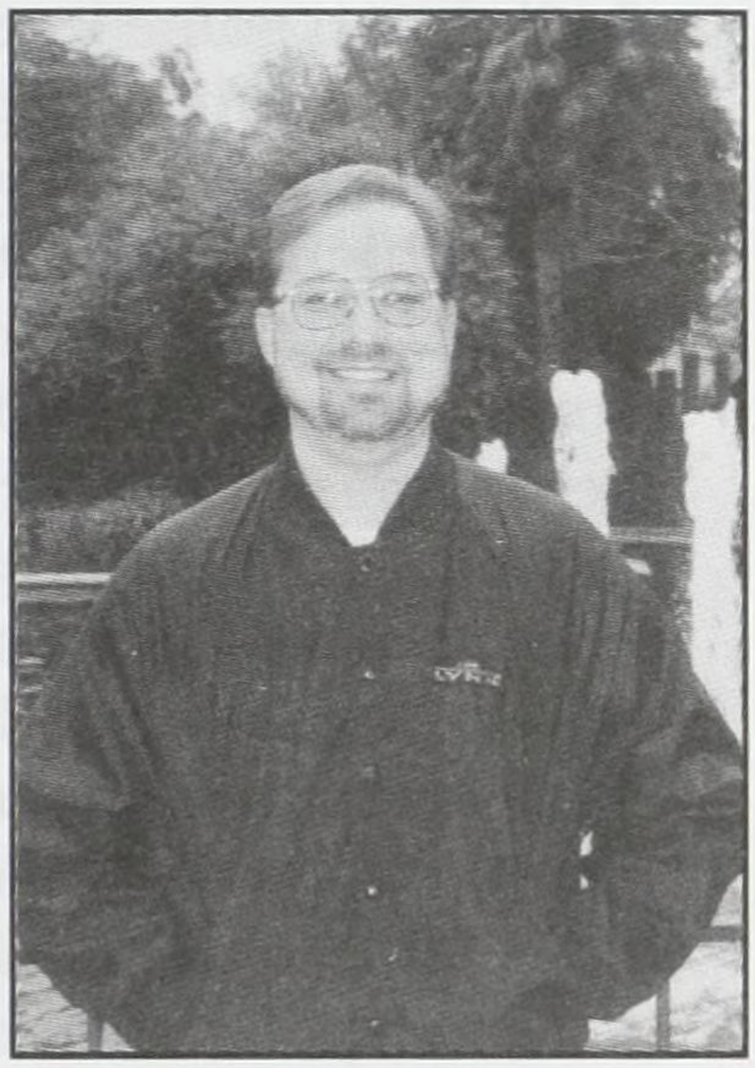
"Probably one of the most important things about game design is that you can't let your personal whims get in the way. What you want may not always be best for the mass public. Designers have to create something that most people like and want. Be open to criticism, especially when you actually get something in the works and people either like it or dislike it. If you get a lot of people who dislike it, pay careful attention to that feeling. That's hard to do for a lot of people. If someone who has a tremendous ego is involved in a project, I don't care how good they are, get them off."

Jeff sees game design as an art form in much the same way as film, only more so. "It's the equivalent of writing a script or directing—actually it's more of a combination of the two—so in that respect it's completely visual expression. In defense of the industry, we have to take the process one step further, creating not just characters people can relate to or use, but can interact with. We have to come up with every scenario, every possible situation that allows them to act and react with that given character. That's a pain in the neck."

John Skruch

For his entire interactive entertainment career, John Skruch has been with Atari, and he has seen the highs and the lows. But he never planned on getting into the business. In fact, he was a business major with another degree in mechanical design, and for thirteen years he designed cannery equipment. "I designed a machine that puts ball-point pens together. And, if you ever eat a breadstick in the United States, it was processed partially by a machine that I designed."

John was led into game design through his boss at the consulting engineering firm. "My boss went to Atari because Atari was teaming with American Express for a high-tech credit card reader/transfer information system. The Atari-American Express deal fell through, but he became head of product engineering for



John Skruch

software at Atari in the home computer division. He dragged me over here in 1982, and I've been here ever since. I started out in computer software, being responsible for the manufacturing. . . when the game was designed and went to get produced, my department owned it." John had to answer, How do you make cartridges? How do you test the product? How do you take a new game and get it produced? How do you redesign the packaging? We actually designed a couple factories for reproducing software. Then all the poop hit the fan here at Atari. There were literally thousands of people a week getting laid off. I became operations manager for Atari Soft, a small renegade group of fifteen people in the marketing departments of the Atari home computer division. We all contributed some ideas and designed some of the Atari Soft product. When the company was sold in '84, I was one of four people kept in a four-hundred-person building. They said, 'This guy's got a technical background, keep him around.'

"Lately, I have gotten more and more involved in product design, not only in game software design—things like disk-operating systems and telecommunication software for an 8-bit nesting computer line. But the games are always my favorite, and always will be.

"So for the last thirteen years as of June, I have been involved in one way or another with software with Atari. I just kept getting more and more involved in the designing side of things. You're trained to create something that, in general, hasn't been done before, so I found it kind of natural both from my early training and from my musical background—for a few years, I was a full-time bass guitar musician in a rock band. We played Tahoe and Vegas and Reno. And between the design and project management aspects, my engineering background, and creating songs and arrangements, I found that's been a good basis for designing a game product."

John sees a lot of similarities between music and games. "They actually do the same thing. You're touching people's innermost

feelings on some level, so the hit songs and the hit games are the ones that touch the person's heart strings in some fashion. There are comedic ways to do it. I did a game called *Kung Food on the Linx*, which is a fighting game where you turn into a little gelatin guy who fights a whole bunch of mutant vegetables in the refrigerator. It was just a food-fight thing with a sense of humor. That neat sense of humor also touched a deeper level, like a kid playing with food. Music does that also."

John also finds game design fulfilling in much the same way music is fulfilling. "I'm more of an arranger. I have a strong sense of composition, arrangement, and I really like to have a product be very polished. I really don't like jars and jangles—I like a good flow, for people to go, 'Wow, that's really put together nicely.'"

His chief frustration echoes a familiar theme, especially in the cartridge business: coming up against technical limitations. "You've got this great idea—a vision of how this particular aspect is going to work—firmly in your mind and you don't have the room on the cart. With a CD, you don't have that problem, but in the old days, especially with the 2600 when you had a 4K cartridge size and 28 bytes of memory. The limitations of the hardware didn't always fulfill the visions of the designer."

John has found that as a designer it's good to know at least a little about a lot of things. "In game design, there's a technical side and a creative side. On the technical side, you do need some knowledge of engineering. Mine happens to be mechanical, but I've picked up quite a bit of programming expertise. I try not to let that knowledge get in my way, because if you can see something in your mind, express it clearly, and challenge somebody to accomplish your vision—like the programming team, or whomever is working on this particular project—sometimes they can surprise themselves and are able to do something they initially thought was technically impossible. So, you need to know the rules so that you know what you're breaking.

"On the creative side, get as much experience in as broad of a range as possible because games are visual, games are oral, and games are physical. The more broad-based knowledge you have, the more you can bring to the party as far as product input. So people should try a lot of different things: filmmaking, playing a musical instrument, white-water rafting, and so on. There's all

of these different kinds of experiences that will show up in one way or the other in the products you're designing."

Good interpersonal skills are also extremely important in John's experience. "Sometimes it's a group of people with the single mind working toward a single goal, or it's a group of people all having different ideas and engaging in cat fights. Those are the two extremes, and in general, it tends to be somewhere in between. When you have a product leader for a given product, that person needs to maintain the main vision. When it starts getting sidetracked, that person has to make a tremendous amount of juggling decisions in a product. There may be budgetary limitations, technical constraints, time constraints, hardware limitations, or what have you. No two projects are the same.

"It's not like, 'Well, I'm going to design game A today, and tomorrow I'm going to design game B, following the exact same steps.' Every game is different and brings different problems and solutions. When you're working with highly creative people, they tend to be very individualistic. The tough part is the producer's job—keeping everyone focused on the task at hand and on that common vision.

"Being able to bring diverse people together is rough. I remember working on one project where the two programmers ended up not talking to each other. I literally had to get them into a room and play 'daddy' and get them to kiss and make up. It actually was kind of like an *I Love Lucy* episode, because one guy would say, 'Well, you tell him because I am not speaking to him.' They're both in the same room, but it was one of those situations. So I was father confessor, daddy, babysitter, peacemaker, what have you, to get them to even speak to each other to finish the project. It was pretty berserk. That's why I said sometimes it's like a cat fight.

"Other times, you get cooking. It's like when you're playing music, and you get into an extended solo. You go off in a direction and the entire band goes off with you. You're flying in unknown territory and you're all together. It's really an exhilarating experience. You can do the same thing in game design. The game is going along okay, but you're not really incredibly enthusiastic for whatever reason. Then all of a sudden you hit on a hook, a neat new aspect that makes the game something different, some-

thing special, and you all fly down and make that particular aspect happen—that's real neat."

John's next project is not only slated for Atari's vaunted next generation game system, the *Jaguar*, but it is an update of a decade-old arcade classic, *Tempest*. When asked about the design of *Tempest 2000*, John becomes even more loquacious and modest than usual. "First, we took a classic game and said, 'Okay, we've never done *Tempest* on another platform.' We decided that this is really a great game with really nice play-mechanics. But what can we do to bring it into the twentieth century on a brand new piece of hardware? What can the *Jaguar* do really well that compliments the reasonably simple play-mechanics of *Tempest*, bring out its best essence, plus take *Tempest* farther than when it was done in 1981? So we were doing things like wanting to make the web solid instead of just wire frame because the old coin-op didn't have effective graphics. So we wanted to fill that in and do some spectacular effects. Our watchwords were 'Why fade it when you could melt it?' We were melting things all over the place.

"We talked with the musicians who were doing the sound track—which was contracted out—about the stuff they came up with. We gave them a copy of the video tape of the play we had at the time, they watched it, got a feel for it, and then got the music to work really well with it. It was all what fits and what works. We were trying a lot of different things with the sound effects. We went through them four different times, for example, trying to find the right sound for the flippers going up the web. We finally digitized the sound of smacking a tall Coke bottle with the round end of a spoon very hard. There were two of us wandering here at Atari, smacking different objects with different things, trying to get just the right sound. People were looking at us like we were crazy, or crazier than usual, but we were hitting Pepsi cans, plastic bottles, pencil sharpeners, and staplers. We were even hitting the microphone against stuff trying to get the right sound. The whole thing blends together so well that there's a lot of stuff you don't necessarily notice. But it was our effort to try and get everything to work just right with each other. I mean, I get really picky about simple things like the sound of the flipper walking on the web, which actually gets pretty drowned out with all the other stuff going on, but it's there, and it's just the right sound."

John sees the future of games and game design heading down a twin path. "There are going to be two types of games, but I think they'll both be very successful. One type is going to be closer to reality than ever before, because with the advent of inexpensive and very powerful technology you'll be able to have characters sliding through mud and getting dirty, or be able to walk over to part of the background, rip a limb off of a tree, and smack somebody over the head with it, as a very, very realistic experience. The second type has never been done before. There's going to be different play-mechanics and environments never before thought of or seen because you've got reality and a simulation of reality—and then you've got cyberspace, which is totally fantastic. People would like to go where they've never been, but that looks like someplace they've been before. It will either be very real or very unreal.

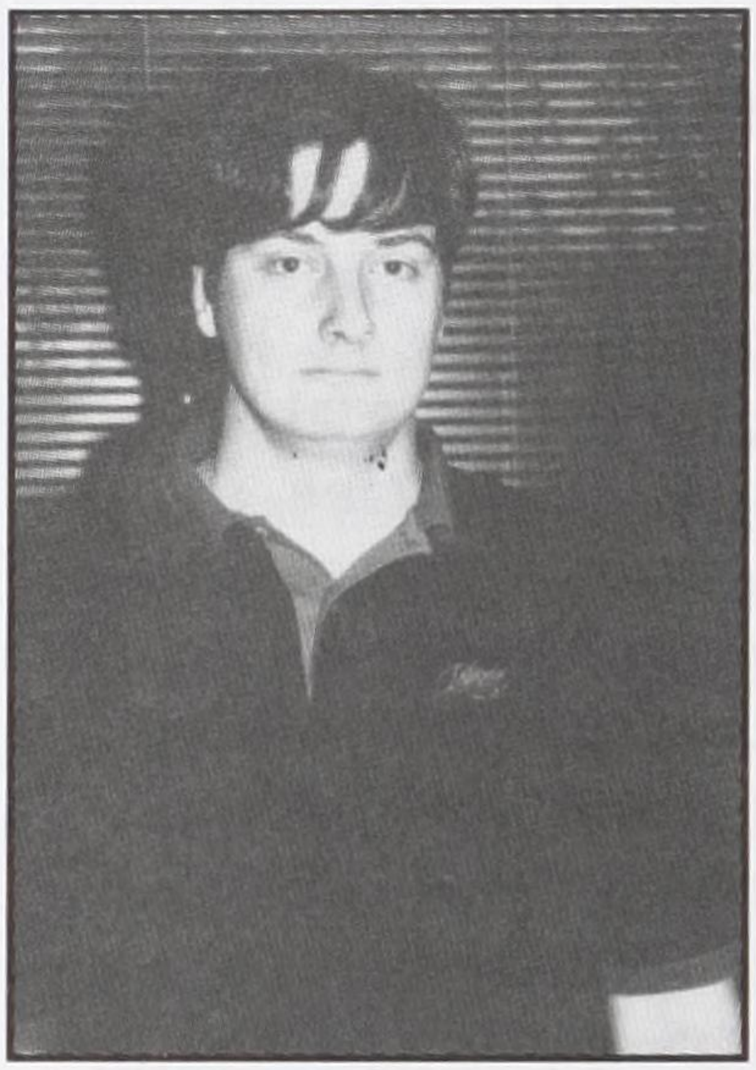
"Play-mechanics tend to fall in groups, and there'll be variations, but I think there's going to be more of the *Pac-Man* phenomena—easy to learn, hard to master games. Then there's the extremely complex and involved 'I'm living in a novel or RPG reality' games."

As a parting thought, John observes that the debate over what defines a product as multimedia is ridiculous. "Any game is a multimedia experience. Really, it's graphics and sounds and interaction. That's interactive multimedia, and it's been that way since the 2600 days."

Jeff Thomas

When Jeff Thomas takes a break from the rigors of game design, his activities can be summed up in one word: games. "Geez, I like to play games, and that's really about it. I play *Street Fighter II* most of the time." His busman's holiday also extends to spending time with his computer. "I do some programming stuff. I play with ideas, I do some writing, I think about future products, and things I have to work on. But I usually spend all my time away from here and at the arcade playing games, and usually playing *Street Fighter II*."

The twenty-two year old, originally from Long Beach, California, joined Interplay Productions in 1991. Prior to that, he was a game tester at Disney Software. His industry credentials include



Jeff Thomas

Claymates, *Clayfighter I* and *II*, and *Clayfighter* for the Sega.

"I was the designer of *Claymates*. I designed all the levels and the characters. With *Clayfighter*, I originally was just scripting all the characters. I took over the game design toward the middle of the project, and then, by the end, became the producer of that project. While the Sega *Clayfighter* uses outside developers, I'm the producer of the product."

Jeff has been interested in game design since he was nine years old. "I started playing games and got really involved in them. Then I started writing game ideas down and started designing them, and it was always something that I wanted to do. I finally got the opportunity to do it, so I jumped on it."

And what does Jeff like most about his dream job? "I think it is actually being complimented on your work. When you do something that's original—and it comes across the way you want it—and other people see it and think you did a good job, it makes you feel good. Just having an idea come through the way you want, and having other people impressed by what you did. *Clayfighter* was my first opportunity to make a fighting game, and since I was a big *Street Fighter II* fan, that was like the thing. I was totally excited about doing it so that would probably be it."

Jeff has very little trouble working with other people or working long hours. The only thing he doesn't like is interruptions. "If I get bugged every day, it's hard for me to get back into the beat, or I'll have a bunch of stuff I'll have to do, or I end up forgetting one thing, and then remembering it, so I have to bolt out to do it again. I hardly ever work at home. I try to do that as little as possible. I like being in the office working around the people I work on the project with, and I usually stay here twelve hours a day—maybe on the weekends and holidays and all of that good stuff—because I like my job too much."

Entering game design should only be done by people "who have desire to do it. I mean, they have to know what they want to do. People can't say, 'Well, that would be cool because that sounds

like an easy job to do.' That's not going to work. They have to have the desire to do it, like I did. I was playing games—getting really into them—and started learning about computers. I tried to figure out how they were made, and started coming up with neat ideas about what was new, different, and out there.

"If you have that ability, you can go in and look for a job in a quality assurance department. From there you're able to get opportunities to show how creative you are and that you have the desire. That's what I did, and that's how I ended up moving up in the company. It would probably be the same for anybody. They wouldn't jump right into being a producer, or designer, but they would come in at the beginning and learn from the bottom up. It takes a little bit more time, but it does pay off, if you work hard."

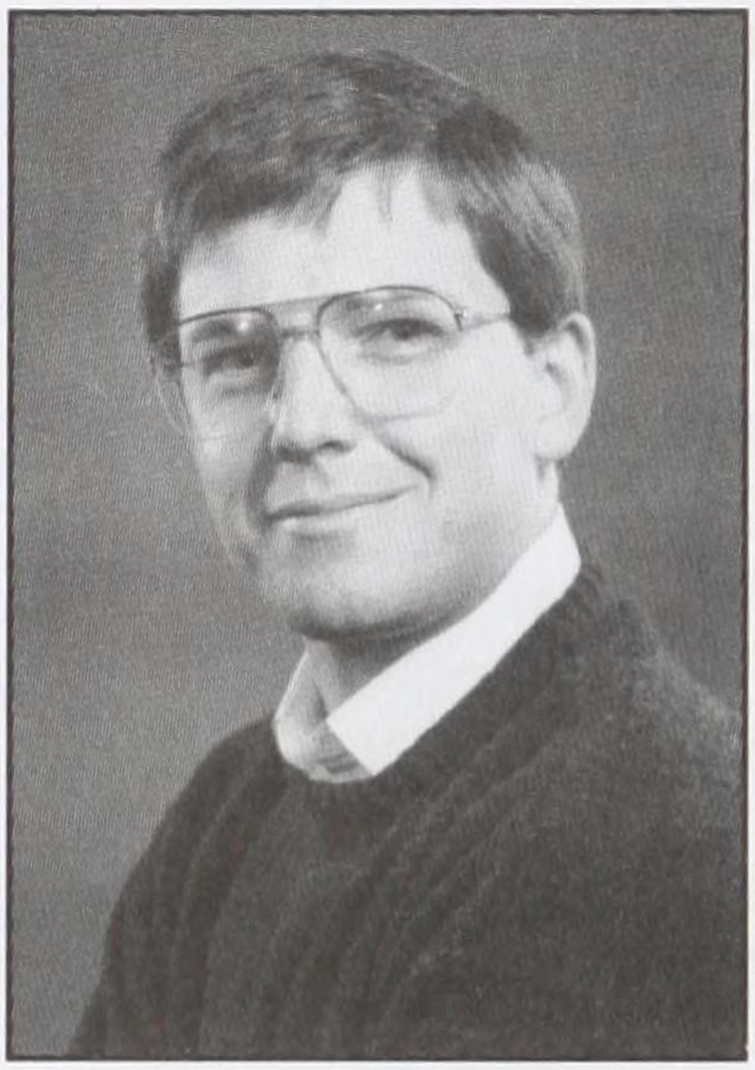
While Jeff attended two years of college, he says, "It was the most wasted time I've ever spent. I actually learned more in high school. There are classes out there that will probably teach you stuff you're supposed to know for doing specific jobs. But there are no classes you can take to do what I do, or what a lot of other people around here do. It's just hands-on practice.

"Designing is really an art form. I think people who make games are taken for granted, and people think, 'Gee, that's all you do? That's so easy.' But it's really not as easy as it seems. I was really surprised about what all was involved. When I started I thought it was really simple, but the more you get into it, the harder it gets. You have to be really creative, you have to have something that sets your product apart from what's out there, so it does take a lot of ability and a lot of creativity."

Creating something that is unique can take time, sometimes much longer than even the professionals estimate. "I'm in that situation now: I am trying to wrap up a product and it takes a lot of work, and the fun turns into a lot of long nights. But, the gratification of finishing is well worth it: You get a week of rest and enjoyment, and you can't help but want to do it again. At least I'm ready to jump back into the next product deadline and go for it."

Jeff Tunnell

Industry veteran Jeff Tunnell purchased his first computer in 1980. After playing games on his new Apple, he thought creating games would be a great way to make a living. "When I was in



Jeff Tunnell

college, I was a musician, and it seemed like a lot of people who were into music wanted to make it big in music. But music is just so huge that there's so much luck involved. Game design looked like a way to have creativity and have more control over it." As the creator of almost two dozen titles, Jeff Tunnell is best known for *The Incredible Machine* series and *Heart of China*. The former is a series of Rube Goldberg-like kinetic puzzles, while the latter is an innovative, state-of-the-art adventure. His adventure games were created under the Dynamix banner, where he served as the head man before moving on to create his own development house.

Jeff Tunnell is one of the most creative people, and original thinkers, in the electronic gaming industry. He strives to maintain high standards of product, while fostering a working atmosphere of serenity. Most importantly, in his own mind, he knows where he and the industry he loves has been, and has glimpses of their intertwined futures.

In light of his experiences, he has sound advice for anyone wanting to enter game design. "Make it worthwhile. Do things that exercise your imagination and creativity as much as possible: Take writing and art classes so you can communicate your ideas to people. The development team, the corporate marketing department, have to buy your product, so you have to be able to tell them what it is."

Jeff's typical day is now actually more involved with producing than designing. "Really good ideas don't come at getting up at eight in the morning, going to my little design cube, and saying 'You're going to design now.' You can do that on interfaces once you got the big idea, but ideas like *The Incredible Machine* come over a long time. So now I might have my plate of ten or twenty really good ideas that I can do, but then there's the plate of two or three good ideas that make any sense at one time, so a lot of decisions have to be made: Do I have the right team and programmers available; will a company buy the game right now; is it the right time to publish this idea; is something similar on the

market already; and so on. That's what takes up the most time right now, not the actual idea of doing the design.

"But on the days that I do design, I usually stay home and don't come into the office. I get up, have a cup of coffee, go into my home office, and just think about what are we going to do with this product on the 3DO—how can we take advantage of the technology? That's a fairly straightforward daily process. You just kind of sit there and think. In fact, you might just sit there with a blank piece of paper and think most of the day—write down a few ideas, talk to some people, and go back to make more progress, until you finally write a straightforward design document. But the big 'Ah-Hah!' ideas do come. My most memorable one came on a press tour for *Rise of the Dragon* several years ago. I was flying around, sort of half-asleep and half-awake, and I just had this really great idea for a product. I got up, wrote it down, and then just sat and meditated on it for a while. I think through possible problems and solutions, and that kind of thing. Finally, it's the right time to present the product and bring it to market.

"You just say, 'Well, we're going to do *The Incredible Machine 2*,' and on those kinds of things, no one person has all the best ideas. Usually what I do is just sit down with a bunch of design goals for the product, and say, 'We need some more new puzzles and so many new parts,' or, 'We're having a lot of success with kids and teachers, so we're going to add such and such features to it, like a talking professor who will talk you through the puzzles.'

"Those are the kinds of things that are natural evolutions of the product. Once the breakthrough product comes on the market, you get a lot of feedback from 150,000 or 200,000 users and the marketing department. *The Incredible Machine* isn't really the property of any one person now. I can sit here and think through strategic things to put into it, and great design ideas come up from the different people who work on the team. We just kind of log 'em and say, 'Oh yeah, create a background with mountains, clouds, and the sun in the sky.' They're not interacting parts, but they really make the thing look great. My job is to listen to all the different aspects and get those into the product."

Jeff finds that his true fulfillment in game design comes from the kids. "The absolutely best part is when you do something that you think is really good, but until you watch a bunch of kids play it, you don't know it. Fulfillment comes when you see that

they're having great fun with your product or that it's helping people. It's not really the aspect of design, it's the end results of the design that's the neatest part of it."

Tunnell is no stranger to frustration, though. "The hardest part is doing something that you know is absolutely wonderful, like *The Incredible Machine*, and you can't sell enough product because the sales force goes, 'Geez, what is this? I don't know how to sell this.' So you have to fight for what you know is good. For example, while *TIM* is really gathering momentum now, if we had gone off the first couple month's sales, it would have been a dead product now. It didn't go flying out the gate, it wasn't a *Mortal Kombat* or anything like that, so I think the hardest part of having a product that you know is great is dying for it and then standing up for it.

"There are no bad parts of the actual design process. What could be bad about it? All the time being creative and having fun. What could be bad about sitting around with, or working with, a team of good creative people? If you're not happy, you screwed up at the start."

Clearly, Jeff enjoys the collaborative process and the recent improvements in technology. "Up until now, for me, working with the sound department has always been secondary because we had substandard music and sound equipment, but with CD-ROM, we've been able to go into the studio, write wonderful music, and use live musicians. That's really brought music up to a level much closer to graphics. Going into the new format is especially exciting right now. Graphics have always been exciting because you can finally bring your ideas to life and say, 'Geez, I didn't know this part was going to be this good.'

"Working with artists has been really exciting, and we're entering a new era now where it's not just one artist. We've always had teams of artists, but it's now way bigger because of CD-ROM, and we need to fill it up or use outside animation to help us. That actually isn't as much fun, but it's exciting right now. My favorite thing is working with the in-house art director, the one who sets the whole look for the game. He creates the icons and their uses, the game parts, and the new characters. It's one of the most fun parts of the product, and is also one of the most creative aspects, where you can just set a parameter and say, 'This is what we need for the design,' and let really creative people

go off the wall and just create wonderful stuff."

Jeff sees game design moving more and more toward movies and film. "These things are so big that not any one person owns it. We have writers who write all the dialog, artists who create the characters, designers who construct the interface, and so it's really, to me, broken up a lot more. It's more of a collaborative team effort, and I hope it will continue to be so."

A fifteen-year veteran, Jeff is seeing the industry blossom into maturity, and he doesn't want anyone stealing gaming's thunder. "We've got all these new people coming in from Hollywood trying to tell us how to do games, as if we don't know anything. They're coming in and trying to reinvent the wheel with interactive movies. Everything we've talked about for the last ten years is suddenly brand new again, and these people act like they have never heard of it. I guess the history of the industry just doesn't count, as if we were too small or dirty. We've got new companies, like Rocket Science, who are all of a sudden going to tell us all how to do it. So it's all new again, like it didn't exist before."

Will Wright

Will Wright changed the computer world with *SimCity*, a program with an ingeniously clever premise: allow the user to construct and run a simulated urban environment. It was a natural, and it started a wave of consumer demand for simulations that continues to this day. Maxis alone has published such entries to the genre as *SimEarth*, *SimFarm*, *SimAnt*, and the recent *SimCity 2000*.

"I've always been fascinated by simulations," says Wright. "I built plastic models of ships and airplanes when I was a kid. After high school, I moved to Greenwich Village in New York and taught myself to program on the Apple II and Commodore 64."

In the early eighties, Wright moved to California to pursue his vocation as a game designer. "I was working on a game where the player bombs a little city," he explains. "I had to set up the city, and I found that was more fun than bombing it. So I started designing a city simulator as a game, but I couldn't find anyone who wanted to publish it. They said if players couldn't win a game, they wouldn't buy it."

Fortunately, Will met Jeff Braun at a barbecue in 1987. "We

decided to cofound a company," Will says, "and develop the game in earnest." The game was, of course, *SimCity*, a little number that has sold approximately 3 million copies worldwide on all platforms. "I'm still surprised it was such a hit," Wright admits. "I had thought that only city planners would buy it."

Will admits there are positive and negative aspects to the game design business. "I like creating complex systems that give people an environment in which they can play and explore. The frustrating part is getting the simulation to run on all the various configurations out there—especially when the game has to be finished by Christmas. This is the first year in a while that I've been able to take Christmas off. It's hard to enjoy Christmas when it's like this big train bearing down on you!"

Wright's current big project is called *SimWorld*, which will eventually link up all the Sim games into a maxi-simulation. "Personally, I'm working on something called Project X. It lets you zoom further into your city, and it will get into simulating human behavior. I've avoided simulating human behavior in the past because it's so tough to do, and everyone has an opinion about how humans behave—so 'errors' are more obvious than if you're simulating a city or world.

"You'll probably see something come out of Project X in 1995. Further in the future, I think it would be fun to create an on-line SimWorld environment, and spend my time tweaking things and seeing what other people are doing."

Rand and Robyn Miller

As the team that created *Myst*, one of the true phenomena of electronic gaming, the Miller brothers have become the most famous multimedia brother act since the Carlston brothers formed Broderbund (interestingly enough, the publisher of *Myst*) in the early eighties.

The story of *Myst* has attained almost legendary status since its release in 1994. Six years earlier, elder sibling Rand was a bank programmer while Robyn studied anthropology. Rand developed an interest in developing children's software. Problem: He wasn't really computer savvy—bank programming is more akin to data entry than any creative process. So he called in his brother, an accomplished artist. Robyn wasn't exactly warm to-

Robyn & Rand Miller

ward the idea of working on a computer, but a short session with a Macintosh convinced him otherwise.

Rand, on the other hand, was a longtime hacker and fan of computer games dating back to the days of mainframe text adventures. "I never thought I'd write games for a living," he confesses, but after a decade of performing data entry and other programming functions in a bank, Robyn turned him onto the Mac. "I instantly saw the lack of high quality software for young kids—I had two daughters at the time. And I thought Robyn, being an artist, would be able to provide the artwork."

In 1988, the duo (under the name Cyan) sold its first "world," *Manhole*. In a sense, *Manhole* and its follow-up, *Cosmic Osmo*, revolutionized the way people thought about entertainment software. Neither programs were games; instead, they were exploratory adventures with plenty for young users to see, hear, and do. There were no "wrong" moves. No judgments were made at all regarding player skill: The product was like a kid's dream cave, a wondrous, half-magical environment in which they could open up their imaginations and learn to interact with a thinking machine.

"From that point on," Rand observes, "our products have been more evolutionary than revolutionary. The worlds became more complicated, and with *Myst*, we aimed at a higher audience, so we put a goal in there, but you still couldn't die." Once the threat of death was eliminated, gamers could feel totally free to experiment, go where they wished, and do what they wanted.

The Cyan world exploration games were so successful with youngsters that the Millers next produced *Spelunx* and the *Caves of Mr. Pseudo*, a sort of *Manhole* with more educational content. Obviously, however, the format the Millers had created held fascination even for adults, who often found themselves exploring *Cosmic Osmo* and other Cyan creations for embarrassing lengths of time. The next step, therefore, was a game that used the explo-

ration format pioneered in these games, and raise the experience level to a point where adults would be challenged.

The result was *Myst*, perhaps the first ever mass-market cult hit. As with the Star Trek television and movie series, *Myst* has both a mainstream following and a hardcore fandom who consider the program everything from the ultimate VR experience to a quasi-religious revelation. Put another way, *Myst* features enough eye candy and brain-busting puzzles to keep the average computerist fascinated, while simultaneously maintaining a mysterious quality that suggests much deeper levels of meaning.

"We started out trying to make a storybook 'with Manhole' and it ended up being a world, much like *Myst* is—a point and click environment. Of course, it looked a lot different," Robyn recalls with a laugh. "But when I saw how relatively easy it was to put something like that together, how fulfilling it was, that was an amazing thing. I think it would be amazing for any artist; in a way, I wish more artists would do this, instead of just creating one image or even moving images. Creating interactive worlds composed of your images is very fulfilling.

"The only frustration I ever encounter is doing work by hand. Now that's just the kind of artist I am; there are a lot of artists who excel at hand drawing. But for me, the only frustrating thing about working on computer is when I start running out of the power I think I should have. When things slow down. But that's nothing compared to the frustration I have when I have to do a detailed, photo-realistic painting by hand. When I'm working on the computer, it allows me to conceptualize the idea, to lay out what I want the idea to look like. The computer does all the busy work—figuring out where all the shadows go and how the object looks from every perspective. With computers, it's just so fast, you can have an idea, create the environment, and the next day move on to someplace else."

Although he works best in isolated places without distractions, Robyn admits that his preferences sometimes conflict with the nature of his chosen field. "It's getting harder and harder because, as an artist, artists generally prefer to work alone, in an empty studio. But the kind of work we're doing demands that we communicate a lot and that we are constantly aware of what other people are doing. It may make me less efficient, but it makes the team as a whole more efficient.

"Working with other artists, however, is very enjoyable. Since we've brought in other artists it's been enjoyable because we challenge each other. I'll look at what someone else has come up with on a screen and I'll be amazed, thinking, 'Wow, how did that person come up with that idea?' And that encourages me to come up with a cool idea for whatever I'm doing. It's almost the competition where we are encouraged to do more than we might otherwise. As long as the competition remains friendly.

"As far as working with the programmers, I don't work with them as much. I don't sit down with them on a daily basis."

Robyn peers into his crystal ball and sees "the technology allowing computer games to become more advanced. In my own view, I think the ideal is for things to become more and more realistic. For example, in *Myst*, when you're walking through the forest everything is kind of static. Ideally, trees would sway and animals would be running around. When we have the technology to both render and display all that—the more movement we can get on screen, things like clouds moving through the sky—it will give us worlds more like the real world."

At the time of the interview, the brothers were hard at work on the sequel to *Myst*. Unlike other creative teams we spoke to, however, the Millers (especially Rand) find the early stages the most difficult. "It's very frustrating," Rand admits. "It's creative work, and creative work is very hard work. You spend a lot of time in brainstorming, just throwing out ideas. I think it's satisfying when it works, and it's frustrating when it doesn't. Of course, if you'd caught me in the cycle of actually constructing the software, I might've said the same thing about that. Still, it's very satisfying when you see ideas that you've worked on for so long finally taking shape and actually being able to walk around these worlds."

The most satisfying part of the process? "The actual first few months of shipping with nice reviews coming in and good sales, I'd say that was the most satisfying part of the process. But there are so many aspects of this work that can give you a nice, satisfied feeling."

Robyn looks into the future and sees diversity. "There are just so many directions software can go. *Myst* is just one aspect. I see, though, overall, the quality will continue to improve. The content will become a larger, more important part of the game.

But those things can be implemented in so many different ways. It's just a matter of more freedom."

Patrick Charpenet and Hubert Chardot

I-Motion's talented duo of Hubert Chardot and Patrick Charpenet embraced one of the great challenges of electronic game design when they set out to frighten gamers in the ambitious, VR-style computer creepshow, *Alone in the Dark* (I-Motion/PC). That they succeeded is testified to by the success of the sequel and the thousands of computer gamers who anxiously anticipate each new edition.

Hubert Chardot, thirty-seven, was born in Lyon, France and entered into the game business through the famous French development house and European publisher, Infogrames. Through I-Motion's business relationship with Infogrames, he soon found himself involved in I-Motion projects as well. Hubert always wanted to write and credits his early education with fostering his love: "I was not a very good student. I preferred to go to the movies, and I had to find very good alibis for my teachers and my parents. It's the best formation for a scenarist when you have to imagine a good and realistic excuse."

Chardot has been the scenarist (and many times the executive producer) on the following Infogrames/I-Motion games: *Shadow of the Comet*; *Alone in the Dark I*, *II*, and *III*; and *Prisoner of Ice*. The term "scenarist" is not used in American game development credits, but it best translates as the concept and story creator.

Hubert's collaborator, Patrick Charpenet, was also born in Lyon. This thirty-three year old's many interests include comic books, science fiction, and movies. He's been involved in electronic game design with Infogrames/I-Motion for four years, and his game credits run from "sets, animations, and close-ups" to game designer and director. He's worked on *Eternam*, *Shadow of the Comet*, *Alone in the Dark II* and *III*, and *Prisoner of Ice*. Again, the "sets" credit is not in US usage, but it refers to the creation of background graphics, while "close-ups" refers to special continuity art, usually employed in cut scenes.

Hubert Chardot did not set out to become a game designer—but then very few designers do. His early dreams were of the cinema and star-studded premiere galas. So, like many an am-

bitious young writer, he set out to produce film scripts with his equally enthusiastic and inexperienced friends.

"We were young and full of beans," he remembers. "We wanted to be rich and famous, and we imagined Oscars displayed before thousands of our most famous scripts." He began to seriously work in a little theater as an actor ("the worst of the town"), and then moved on to radio. After working in radio and newspapers for four years, he began managing movie theaters for UGC and PATHE. From there, Chardot moved to 20th Century Fox and film distribution.

Chardot moved to games by chance and with thanks to the writer H. P. Lovecraft. In January, 1992, Chardot met Bruno Bonnell, the top manager of Infogrames. Bonnell was looking for a scenarist who could script a new line of games, *Call of Cthulhu*, based on the Lovecraft novels. A great fan of Lovecraft, Chardot's proposed storyline interested Bruno and the staff, and it was bye-bye movies, hello games.

While Chardot entered gaming in a roundabout way, Patrick Charpenet's path was more direct. He first started playing computer games on Thomson MO6 and on Amiga 200 computers. "I was very interested in the depiction of stories and games on this very new media. The thing that surprised me most was the incredible ability to create fun and relaxation on a very serious media that was supposed to be a working tool. I was amazed by the prowess of computer artists and programmers to make incredible games with very limited hardware. I always remarked on the increasing rate of progress in this particular medium."

Patrick, an art director for a communication's company, integrated his interest in computer graphics with his work. "After a few years work on PC, Mac, and Amiga," Charpenet remembers, "I took a leave of absence from my job to look for a more entertaining way to do computer graphics. Advertising and communication began to bore me. I met Infogrames through a friend of mine who introduced me to the development team of Eternam. I did a few tests and finally was hired, first as a computer artist and later as a graphic manager."

Both Hubert Chardot and Patrick Charpenet find fulfillment in their work, Chardot in writing and Charpenet in concept. Chardot observes that "when you begin to write and when you can translate the things you have imagined to paper, that's ful-

filling." Patrick takes this idea further, finding fulfillment through "diving into a universe, creating new characters, exploring new ideas, and looking for documentation." Frustration is not an unknown companion either. Chardot good-naturedly complained that true frustration is when "one year later you see the game and think how good it could be if you wrote it."

Unlike the early days of game design, most games are now designed in teams. Both men enjoy working with other specialists. Chardot believes that "it's always good to work with different people on the game because they have very different points of view. It brings new concepts and new ideas which improve the game." Charpenet adds, "In this line of design you can't work alone anymore. Anyway it's the most fun to work as a team." Of course, blending work styles can occasionally be problematic. "I work best alone," Chardot says. "Alone . . . in the light." On the other hand, Charpenet needs "dead calm" to think, but "as much chaos as possible" to produce.

Neither man is silent on where they see electronic games, and particularly game design, headed. Chardot sees the future tied jointly to technology and Hollywood. "I think that we are very close to Hollywood in the 1900s, but we're waiting for D. W. Griffith and Charlie Chaplin. Game design depends on the progress of techniques." Charpenet is more direct on the future: "Games, and multimedia, are going to break the house. The trick is to be the bulldozer."

They aren't shy about offering advice to prospective designers. Chardot teasingly suggests "go west, young man," but then follows it with more practical advice. "Designers have a lot of things to understand and to learn. We stand on a crossroad between literature, movies, and technology. Recipes do not exist to realize a good game. The better solution is to read, to play games, to go to the movies, and finally to try." Charpenet picks up his codesigner's thread with the observation that "games are becoming very open to every type of creator, as long as the public likes it and as long as it is possible technically, and even if it's not. People, and us, want to be amazed."

The two diverge, however, on the concept of designing as an art form. Chardot thinks that "games will become a real artistic expression when they give deep emotions to the players."

Charpenet, however, states "images, sound, stories, interactivity—if that's not an art form, nothing is."

Like most designers, neither Chardot nor Charpenet maintain set schedules. Chardot says that his schedule "depends on the day's priorities," he explains. "I write, debate with programmers and executives, meet writers, read scripts or books, or try to fax something in correct English. It's a very pleasant job because every day my work is different."

Charpenet concurs that "no two days are the same." He does have a rough schedule that he follows in a more or less desultory fashion. "I begin the day around 9 A.M. checking the messages, faxes, and problems that could occur early in the morning. Then coffee. Next, I organize the day's work according to the week's objectives and emergencies. More coffee. By now, I have project meetings, phone calls, graphic researches, graphic deliveries, and so on. More coffee and/or lunch. Then it's the same as the morning. Coffee. Between 7 and 8:30 P.M., I head home.

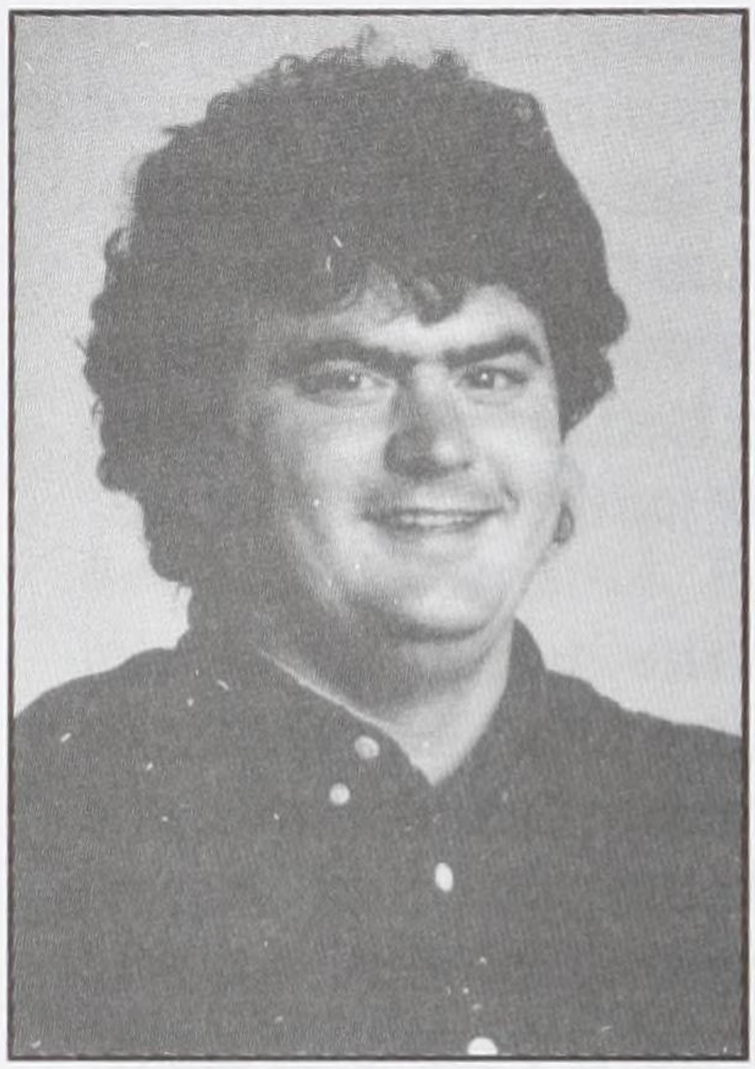
"There, my own computer is waiting for me, ready to start its day's work: exploring new games, graphic finishes of the day, and personal formation on new software. No coffee, although generally the TV or radio are on, as I've got to stay informed."

As electronic entertainment moves into the future, evolving as both an entertainment and art form, more and more traditional artists will find their way into the industry, bringing their traditions, their values . . . and their coffee.

Tim Schafer and Dave Grossman

Though their names may not be as instantly recognizable as some of the other designers interviewed in this book, Tim Schafer and Dave Grossman's work is sure to ring bells. Starting out as assistant designer/programmers for the *Monkey Island* games, they broke new ground, building the foundation for a different kind of adventure with comic overtones and easy interaction. They also served as the project leaders for LucasArts' extremely popular and tremendously influential whacked-out cartoon adventure series (*Maniac Mansion*, *Day of the Tentacle*), carving out new territory in a genre that was threatened with stultification.

"I was first exposed to computer games in the late seventies," Dave admits, "playing the old text game *Adventure* on a main-



Tim Schafer

frame, and I decided that I wanted to write computer games as my career. I then promptly forgot about all that for ten years, until I was dropping out of graduate school at the University of San Diego after working on artificial intelligence and went looking for work, when I stumbled back into the whole thing."

Tim, meanwhile, "wasn't really looking to become a game designer. I was a high school kid pirating software and playing games, and I kind of forgot about it in college. I was studying computer science and getting really bored and depressed. Anyway, I graduated and went looking for work. I was just going back for my second interview for a job with a company that was making police and fire department databases. I was going to work for the section that tracked weed abatement for the town. So I was about to write a weed abatement module for a police database when I interviewed at LucasArts, which I really blew when I mentioned that I had played the Lucas game *Ball Blaster*, which turned out to be the name of the pirated version of *Ball Blazer*, so I almost didn't get the job. Plus I drew pictures all over my résumé, which apparently drew someone's attention. In any case, it wasn't really a career path at all; I didn't really think I'd get a job programming games. Every time I see someone chopping down weeds, I think: 'That could be me!'"

Unlike weed abatement, game design is finally being thought of as an art form, but the notion initially takes both Dave and Tim by surprise. "An art form? Yeah, well, that's interesting; I guess it is sort of an art form," marvels Dave. "It's just another medium, like TV or film, but the one thing that these games have that the other media don't is the ability to respond to the actions of the user in an intelligent and comprehensive way. The interactivity is the key thing; we take a lot from the film and television worlds, but they don't have that ability of the user to reach through the screen. That's something that hasn't been explored."

"I wouldn't want anyone around here to hear me say this," Tim adds, "but I do consider it an art form and ourselves artists. It just seems a little pretentious to say it now because we're not a mature art form. It's such a commercial venture. It starts with a compromise, but it has so much potential. I'm involved with story and characters. Well, you can expose a character in a book through dialog or exposition or action, but how do you expose a character through interaction? That really hasn't been defined yet."

What do they find most rewarding and most frustrating about this new art form, the game creation process? As usual, Dave goes first: "For me, the most fulfilling and fun parts are the writing aspects. I like to come up with a story and put characters in it, write dialog for the characters and flesh them out. I like to come up with puzzles with interesting kinds of solutions that people haven't come up with before. The most frustrating part for me—and something that I'm going through at the moment—is pitching the game idea and getting it built. As you know, games cost a lot of money and it's important to not just have an idea in your head, but you must be able to convince this large group of people that the game should actually be built."

Tim sounds a similar note—his favorite part of the process is the early creative stage. "I like the brainstorming. Like five people sit in a room and meet every day and tell jokes and throw paper airplanes at one another. And everybody thinks you're working. After a while, you come up with a game idea. I also agree with Dave that the most frustrating part is actually trying to get one off the ground. It used to be you could actually sit down and start on a project yourself at home, but now you have to be able to justify spending half a million dollars on a game."

It also involves working as part of a creative team. "The hard part is that they're all crazy," Dave explains succinctly. "We at LucasArts like to hire the best and the brightest, the artists and musicians and such, and sometimes their opinions don't jibe with your own, and they can be really pig-headed about it, just like I can. But on the whole, I love working with all those people because generally they are better at what they do than I am. And as long as I recognize that and let them have their input, then it tends to work pretty well."

"Yeah," Tim agrees, "the secret to making something good is to surround yourself with people who are smarter than you. You

tell them how great they are fifty times a day, keep them motivated, and take them to lunch occasionally."

Where do they see the interactive game media going? "From where we stand, games are getting bigger," Dave declares, sounding surprisingly serious all of a sudden. "As we move into CD-ROM, they're incorporating a lot more imagery and sound, becoming more and more big scale productions, and the budgets are going the same way. In a way, that's unfortunate, since it really squeezes the little guy. It's difficult to get a game done if you're not affiliated with a huge company. We're all making partnerships with film companies. Hollywood seems to be moving in; sweeping in for the kill, somehow, and that scares me a little bit. Things are going to get a lot bigger and a lot more complicated really quickly. But on the other hand, the games are a lot more fun to play. It's great for the end user."

Tim agrees. "There's a lot more pressure to make games more mainstream, games that aren't just for extreme hobbyists. And I don't know what kind of games a regular person might be interested in. Games in general—not just computer games, but board games as well—there are games that only serious gamers play, and others like *Monopoly* and *Trivial Pursuit* that everybody plays. And game designers have to figure out what those 'regular people' games are; what would be the *Pictionary* of computer game design?

"When I design a game I think about it a lot. I try more and more to make it a game that doesn't require specialized knowledge of how a game works. When you play Nintendo-type games you pretty much learn this language that a floating heart means life and a flashing objects mean damage. Now the kids who play the games know what this means, but regular people don't. I like to make games that anyone can look at for the first time and figure out. In *Full Throttle*, I'm trying to make a game that's maybe a little more immediate. Less of a scavenger hunt and more of a physical interaction with the world. With the puzzles, I'm trying to have the goal of being able to just stare at them and figure it out, rather than forcing the player to wander around collecting obscure items and assembling them."

As two of the more successful creators in the interactive gaming business, what advice would the duo offer to would-be game designers?

"Try not to focus yourself too narrowly on computers," warns Dave. "There's a lot more to it than that. Study theater, creative writing, learn about lighting, go to film school—the conjunction of film school and computer people is just a little ways down the road."

"I'm hiring programmers now," adds Tim. "There are a lot of programmers out there, but I had a really hard time finding someone who is both a programmer and a creative person, someone who could actually write, for example. The classes that helped me the most in my work here were not in my major—they were in anthropology and literature."

Bill Crum

Bill Crum became interested in electronic gaming through the nonelectronic version of *Dungeons & Dragons*. "I played the beginning games, and my friends and I started making up our own little games instead of playing the premade scenarios. By the early eighties, I'd gotten into just collecting all the stuff I could find and playing with some of my older friends. When I went to college, I discovered that the more intelligent the group of people that I found, the more games they played.

"I found myself gravitating toward game players, but I didn't have any formal design education or anything like that. It was more a matter of seeing things that I liked, and things that I didn't, so I'd modify the game rules to fit within the systems that I liked. I developed an internal feel for what was and was not so much fun to me."

Crum fell into Westwood's orbit after hearing about a job opening there. He'd spent the previous few years working for a Fortune 500 banking firm, so the change was significant. "I got an old 1982 IBM PC, and made the transition from board and paper games to playing on the computer. I played just about anything I could find that looked like it was a challenge mentally. When I heard about the opening at Westwood, I said, 'Yeah! That sounds like fun.' Brett Sperry and I talked back and forth for about six months. I took the Westwood design test, passed it, and started working on *Lands of Lore*."

Bill attributes some of his success to luck: "I happened to make the right introductions at the right time. When you get to the point of who you know, it gets you in the door. But you have to have

a certain amount of creativity and ability to be flexible. I came in and had my ego bruised a couple of times before I learned that all of what I like isn't necessarily what the mass market likes."

Bill acknowledges that he tends to be a numbers cruncher, which he says works out well within Westwood's setup. "We work together in teams, so what my other teammates may design and what I design may be two different things. But when we put our heads together we end up with a third product that's unique. Phil Goro and I worked quite closely on *Lands of Lore*, and it was amazing to see the things that were part of one idea spur someone else's idea, while toning down the sharp edges of each other's concepts.

"It's neat to see things that you've dreamed up make the progression from an idea off the top of your head through all the hills and valleys of cuts, mergings, and things like that, then make it all the way through and actually end up in the game. It's the little things that make up the grand picture that are most fulfilling."

Bill feels that the frustrating aspects of design are two-fold. He is dismayed by the limitations of the medium. "There's only so much room on a disk or CD, and there's only so much horsepower in home machines, so you can't do everything that you want to do. You have to decide what stays and what goes, and I, like most writers, tend to think of their creations as children. When someone cuts off the arm of your child, well, that's a pretty sad thing."

The second aspect is not having complete control over design. "Your creative control rests, for one reason or another, with a group of people. It's not just a single person going off and creating a game; there are adjustments that need to be made to suit the market, or at least the collective tastes of the group. It can be frustrating whenever a writer has their ideas either criticized or changed."

Bill concedes that he is the type of person who works best when beneath the shadow of a solid deadline. "I get quite a bit done during the in-between periods, but I tend to be most productive when I'm under the gun, and we have to have something done yesterday, though it's very tiring, and after the fact I need a lot of time to wind down. I actually seem to get more done when

it can't possibly be done. I seem to work best when the gun is pressed to my head."

Crum is very effusive toward his coworkers. "I've been overwhelmed by the artists, and I'm including the musicians in that. I have no art talent whatsoever, so I think I have a wide-eyed perspective when it comes to seeing that sort of stuff. The ability to crank out musical creations that don't duplicate somebody else's tunes isn't easy. Psychologically, your brain is going to repeat back patterns that you've already heard.

Bill believes that game design "is going to become a more complex animal." He notes that the number of people on a project has grown from the single garage programmer to teams of "five or ten designers working on it." Crum reluctantly notes that the industry is "going more toward the virtual reality type of thing where somebody can get in and be himself in a futuristic or fantastic environment." He would personally prefer to see the opposite. "I think it looks more appropriate for a game to be a little more fantastic. Digitizing is a tool, not a crutch, but we're essentially seeing movies put into the PC medium."

He suggests that anyone interested in game design as a career "read everything they can get their hands on. Be diverse. Take classes in writing, and learn how to write cohesively. People have to understand what you write and not have to have it explained to them."

When asked about his concepts of game design as an art form, he hedges. "It's misunderstood. It's kind of an evolving art form. It's developing, and some day it will reach its full potential."

As a final, parting thought, Bill reminds game designer wannabes to stick with their dream, even when faced with adversity. "If you're a dreamer," he advises, "never give up."

Rob Landeros

Rob Landeros is cofounder and creative director of Trilobyte, the development house that broke away from Virgin and created the classic *The 7th Guest*. As one-half of the company's driving force, he maintains the artistic excellence of the company's line of multimedia products. However, before reaching this position, Rob spent a year in pre-med, which he left behind to draw underground comics in Berkeley. As his involvement with comics increased,

Landeros moved to Los Angeles to attend art school. There, he worked at Lockheed for a couple of years, while continuing the strange, countercultural odyssey to success as an artist. He continued painting and carving scrimshaw for craft shows—taking the ivory carvings and turning them into jewelry.

When Rob adopted a Commodore Amiga, he found a new and unlimited outlet for his artistic expression as a "pixel pusher." He was offered work on a project for Cinemaware and created the graphics for *Defender of the Crown*. After working on several award-winning products, he was promoted to art director. In that position, he was responsible for the graphic design of *Rocket Ranger*, *King of Chicago*, *TV Sports Football*, and many other groundbreaking Cinemaware products.

In 1988, Rob was hired by Virgin Games as the company's art director. While he was at Virgin, he helped develop the artwork for such titles at *Spirit of Excalibur* and *Lexi-Cross*. Rob met future partner Graeme Devine at Virgin Games, which they left in January, 1991, to form Trilobyte.

Though it is a similar type of game as *The 7th Guest*, Rob has tried to outdo himself with *The 11th Hour* (which was started six months before *The 7th Guest* was released). "We had the luxury to see and hear the feedback of consumers, and we could incorporate some of it into *11th Hour*, and make it more of what *T7G* should have been, from a design point of view and utilizing the improved technology." In the redressing process, "we decided to free up the camera from the first-person point of view, and alter the visuals. It's not a hokey game, but a mature and intelligent game."

The 11th Hour turned out to be bigger, literally, than even Rob expected. "We kept having good ideas, especially as the technology and tools advanced. The game started growing, and evolving, beyond what we originally conceived. We initially planned on 15 frames per second, but with the major improvement of MPEG (a standard employed in generating full-motion video), we rendered everything and boosted up to 30/fps. With *T7G*, we knew it would be big, but we weren't expecting *The 11th Hour* to exceed control."

Rob enjoys working with other team members. "The programmers are the ones whom I understand the least. They're in a different little world than I am. But they all seem to get along with each other, and Graeme tends to be the one who deals with

them. But I enjoy, and it's been a thrill to work with, the creative people—artists, musicians, screenwriters, and novelists."

Landeros believes that interactive entertainment design "will concentrate more on story and characters" as it continues to evolve. "It should give more information, like the novelization of a movie. This is just from my end, I can't claim the future for the industry. As for Trilobyte, I hope we keep redefining the future."

Game design "requires someone who is prone to get in there and figure things out. There's also a need for people to be an interesting personality combination of right and left brain hemispheres; both the very technical and the very creative. It's a challenge to find the right people, since you don't usually just hire someone off the street."

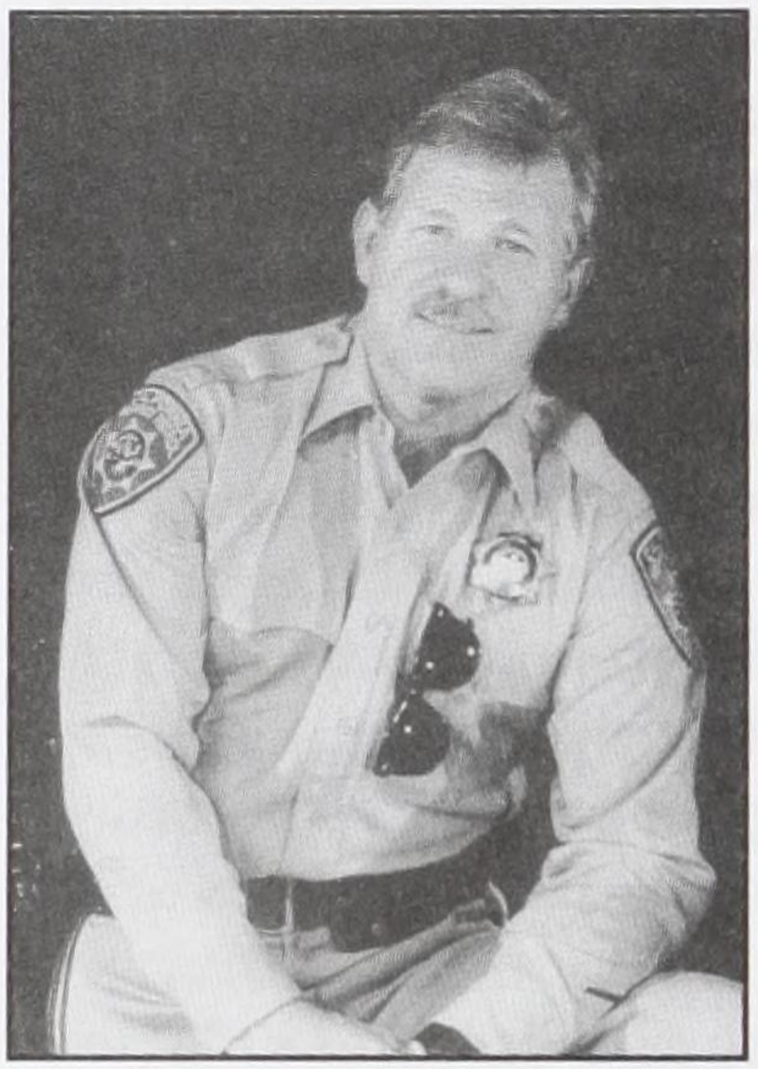
Rob concedes that it is hard to break into the business. "The best bet is to have some sort of talent—artist, storyteller, game tester—that will let you in at any level. The cream always rises to the top. People also need to understand what the company is about that they're trying to hire into."

"Game design is a new art form," Rob asserts, "and we're inventing it as we go along. It may not be anything in the end, but it's very interdisciplinary at the moment. I can't imagine anything more challenging."

Jim Walls

Like many others in the profession, Jim Walls never intended to become a game designer. A fifteen-year veteran of the California Highway Patrol, he was involved in a shoot-out, which altered his career. While the department evaluated his condition for return to active duty, Walls had little to do. "All you can do is sit home because you are still a liability of the highway patrol, so I was just sitting around twiddling my thumbs, and playing racquetball a lot. I met Ken Williams, who invited me to a game of racquetball, and I kept playing with him. He was interested in doing a police adventure game, and said, 'Hey, do you think you could write a story?' And I decided to give it my best shot. He liked it, and that's how I got going."

Walls laughs. "Actually, when I started, I had no knowledge of computers whatsoever. I had no idea that in a year's time I'd be designing adventure games. It's just one of those things



Jim Walls

in life. But basically, I'm still just a cop at heart."

Jim's design secrets for *Police Quest I*, *II*, and *III* (Sierra On-Line/PC), as well as *Blue Force: Next of Kin* (Tsunami/PC), are very straightforward. "I put what I've done into the games. You know, the reality of procedures and stuff with adventure. I like to make the gamer actually become a police officer when they're playing the game." Jim's other game, *Codename: Iceman* (Sierra On-Line/PC) is a military covert operation that takes place, in part, on board a submarine.

One of his current projects, tentatively titled *Code Red*, places the gamer in the role of a SWAT team officer facing down a bunch of radicals who seize a riverboat and hold all the passengers hostage. The gamer must stop the terrorists from blowing up not only the boat but also a dam. "This should be a pretty good one," Walls believes. "We're using high-res art and nice looking backgrounds. The character animation has the darker, super-hero look, like the Shadow or Superman. It looks really good, and I'm excited about it.

"The artists are probably about 75 percent done with the graphics. We're also building an in-house 3-D engine to go along with it at the same time, so it's gonna have a different appearance. But it's going to look good." Jim has yet another plan on the drawing board, called *Murder One*, which puts the player in the role of a homicide detective dealing with the seamy side of life in a large city.

Walls finds that fulfillment comes from improving each new title. "I'm constantly looking to make games better and more fun. I just try to stay up to date with what's possible, and that's more and more and more all the time. I'm trying to make it like an interactive movie where you're controlling the characters. With the high-quality artwork and realistic music, you can do that stuff. The next thing is virtual reality."

Jim pauses and reflects. "What I love is when I get a letter from a fan after the product is shipped. I get letters from kids, who are fifteen or sixteen years old, who say that, after playing the game,

they want to start a career in police work. When I get those kind of letters, then I know I've done a pretty good job."

Jim's business partner, Cheri Loyd, has been involved with all of his projects since *Quest II*. "We've been friends from that point, and she's helped me with at least three stories. It's been real good because I'm fairly hard-headed, and I tend to only see things one way. She can come through with some ideas, and she convinces me of others. She looks at everything through a different perspective than I do. In my mind it's strictly thinking like a police officer all of the time, and she puts a softer touch on some of the characters.

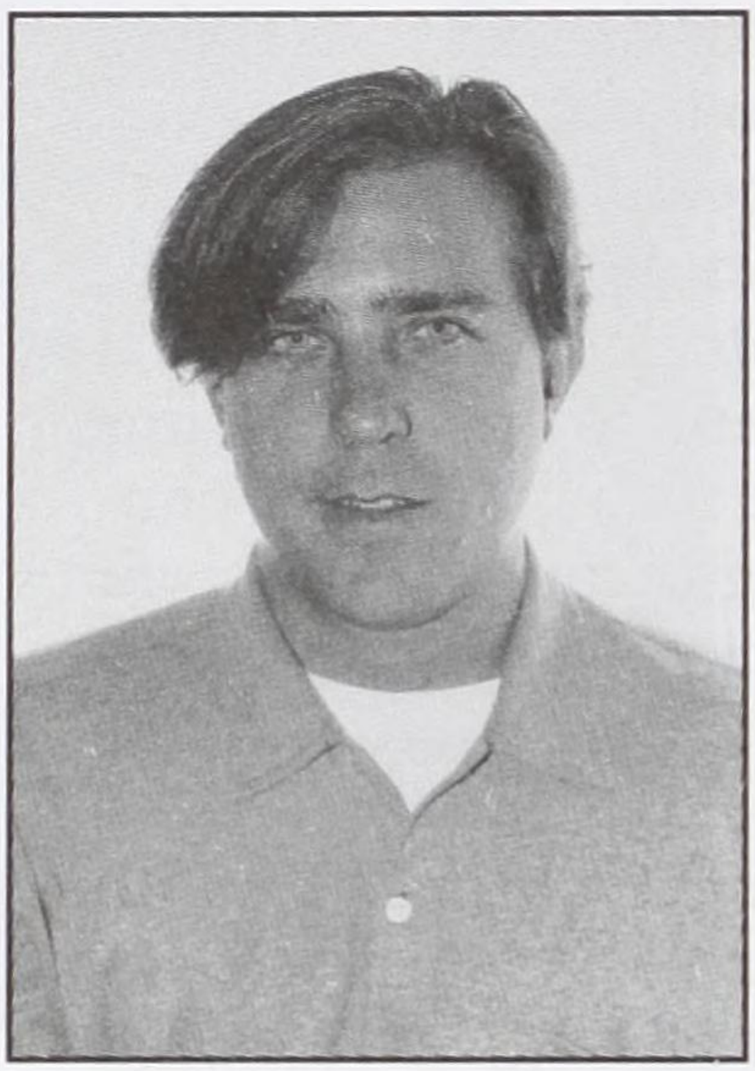
"It's a real experience, too. Sometimes we get a little loud with each other because of ideas clashing and hard-headedness. Usually, I don't want to budge, so it gets real interesting, but it works out very nicely."

Jim sees future game design as involving more people with greater input. "Even now, I come in with a basic design, and it seems to work if you sit down with your team and go over each area of the game, because there are other people that have some tremendous ideas that they can interject. I think that once you nail down the basic design from point A to point B, more people are interested in looking at, and going over it, with you. This way, you can flesh it out to make a better product."

Walls doesn't advocate anyone entering game design the way he did. "For one thing, I am not a natural-born writer. People have to have the ability not only to tell stories well but to communicate well on paper, so do whatever it takes to reach that level. You also have to live the story—what you're writing about—every day. At least I do. I have to be walking, talking, and thinking the story all the time to get ideas and make them work. You have to put your soul into the story, to make other people feel what you're feeling."

Ted Barnett, Keith Kirby, and Rob Fulop

Keith Kirby is the programmer, Ted Barnett is the producer, and Rob Fulop is the director of creative development: these three make up the heart of PF. Magic's star design team. The three bring diverse backgrounds to games and game design.



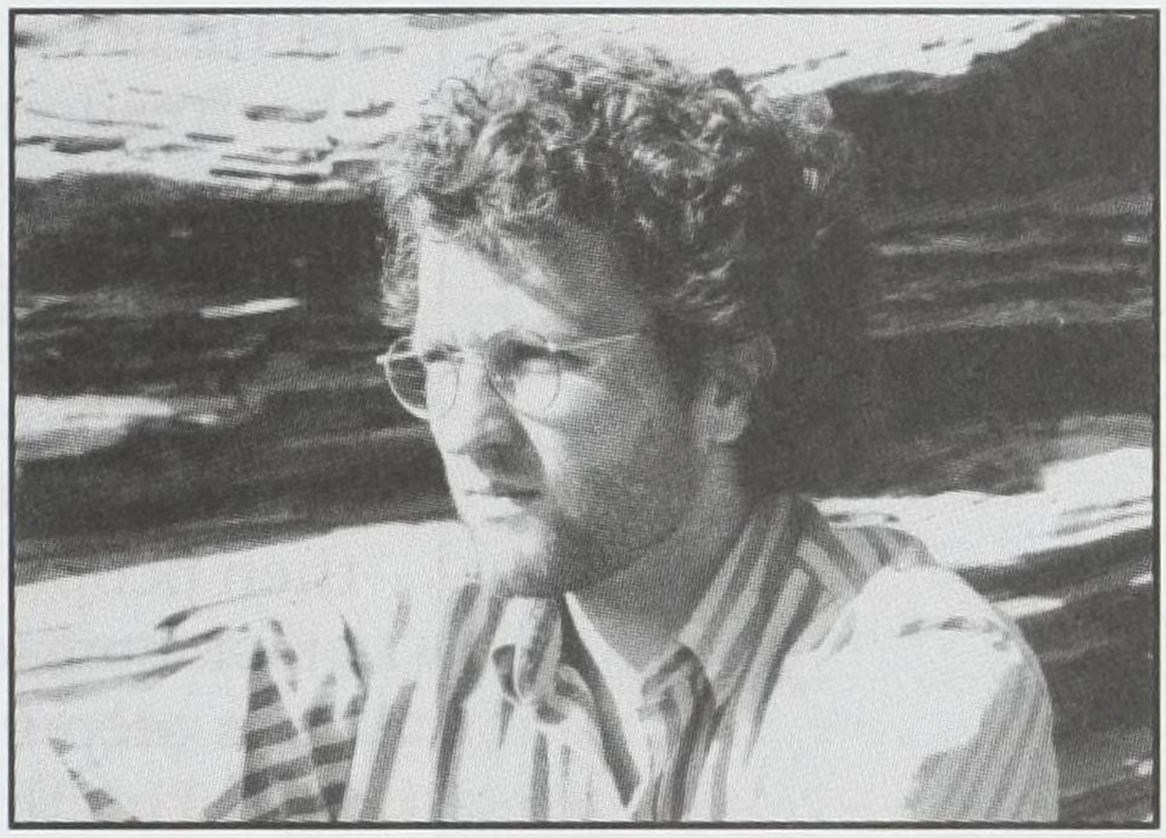
Keith Kirby

Keith, who at twenty-nine has spent six years in the industry, became interested in game design as a result of a serious addiction to video games. "I grew up in a very large suburb that didn't have much stuff for young kids to do. Most of our time was spent at the mall, bowling alley, or any other video game resting area. All of our time was spent playing games. I fell in love, but I didn't ever think I would have the opportunity to do game design. I just kept on trying, earned a degree from the University of California at Santa Barbara in computer science, and took classes in film study, psychology, child development, and so on. I ended up being very lucky and getting a job with Activision within a few years.

"Initially, I couldn't get a job in the game industry—it's a catch-22: You can't get a job unless you have experience. Programmers have highly specialized skill sets and knowledge about the individual game sheets themselves. So I worked at Xerox for a year on Windows programming and then resubmitted my application to over a dozen game companies. At the time, Activision was going through major problems: People were quitting, the company was going down the tubes, and I didn't know better. I got a job out of pure chance because no one else was interviewing. I got a chance to start on a Super Nintendo project with a couple of really great guys, and I spent a year creating SNES titles."

Ted, on the other hand, fell in love with computers. "In high school, I had an Atari 800, and I used to write games in BASIC and a 6502 assembler for friends. I earned a computer engineering degree at the University of Michigan, and when I graduated in '84, I thought about getting into games, but the industry looked like a disaster at the time. So I sold my soul and did business stuff—consulting, Harvard Business School—until 1989, when I joined Apple Computer as a product manager.

"I did some consumer product stuff, including a project—similar to the 3DO—that Apple shelved. After a few years there, and at other companies, I decided I just wanted to get the heck out of hardware and work on my favorite type of software: games. I



Ted Barnett

came to PF. Magic, was immediately paired with Keith, and that's where I am now."

Rob Fulop was a professional game player. "I dropped out of college in 1978, played poker and backgammon for a living, and then I turned to Atari. It was just starting, so anyone with a basic idea of programming could get a job. I was always a game player—card and board—but video games hadn't been marketed yet." Once they were, however, Rob made his mark, creating the phenomenally successful *Demon Attack* on the Atari 2600 as well as several other successful titles for Imagic. He has had experience in virtually all sectors of the gaming world—he even created *Rabbit Jack's Casino*, one of the most popular features on Quantum Link (an earlier incarnation of America On-Line).

Even with their diverse backgrounds, the trio finds the same things satisfying. Rob notes: "I find watching a kid playing PF. Magic games to be very fulfilling. I think designers get a kick out of seeing their intended audience enjoying the fruits of their work." Keith agrees that this is "the number-one motivating factor. I got into game design because I wanted to make the games I wanted to play." Ted concurs, but adds, "since the industry is young and everything is similar, the most fulfilling thing is inventing surprises that people have not seen before."

Then Keith suddenly remembers, "I forgot one other element, game programmers get the girls."

All three agree that they have an excellent working environment at PF. Magic and the opportunity to customize their schedules. Keith observes that "we have zero restriction on what hours we work, freedom, and high motivation behind the products that we create because of that. We really feel that what we are doing we're doing for us and not a paycheck, which is important."

Ted aims for "normal hours, until getting into a real push forces me to put in late hours. I just do best when there is pressure and time restraints. As producer, you assume the best of your team,

Rob Fulop

let them do what they want, and give them the freedom to create things." Rob adds: "You're talking to our 'A' team, so we set up an environment conducive to doing a great job. There is an internal discipline: It's not like working at the fun house. As long as stuff is getting done, and it's good, then people can set the tone, but if projects start coming in late, then management does step in."

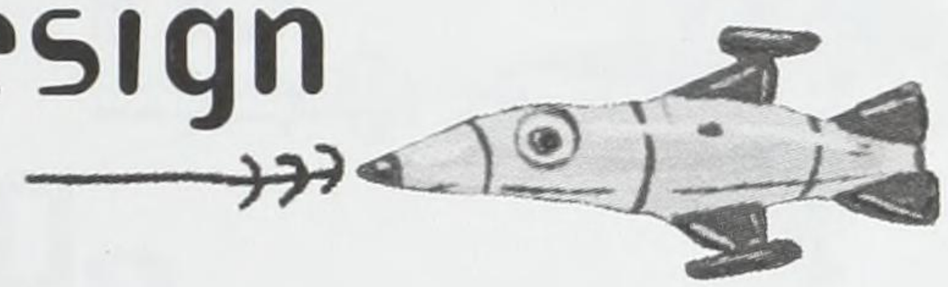
The threesome has no problem coming up with the positive aspects of working with other professionals. The consensus is that if they are good and contribute ideas and pay attention to details, then the designers can focus on game play and their details. Keith elaborates, "The best thing is to show up one day and have a list of new things, or a piece of code that integrates with and enhances the product. In *Ballz*, cool sound effects was just the icing on the cake, since the sound track was really funny. But Mark Miller at Neuromantic Studios did some really cool sounds and the game was 30 percent better."

There is dissension over whether game design is indeed an art form. Keith plays devil's advocate. "I don't think this is art—it's entertainment. The word 'art' has been generalized too much. I want people to have fun, be addicted, and miss dinner." Ted, however, feels differently. "I think that Keith is being a little simple. *Ballz* is an example of a brilliant design, and the theory of form following function. *Ballz* is designed around the limitation of the Sega Genesis, and I believe it is an art form."

Keith and Ted's parting advice to potential game designers is: "If you think this is worth any amount of trouble to accomplish, then get into it. It is a lot of fun, and the industry is only going to get bigger."

PART FIVE

Careers in Game Design



You know the history. You know games. You've even got a pageful of brilliant new ideas that will one day set the gaming world on its ear. Now how do you get a job? This is the theme of Part Five.

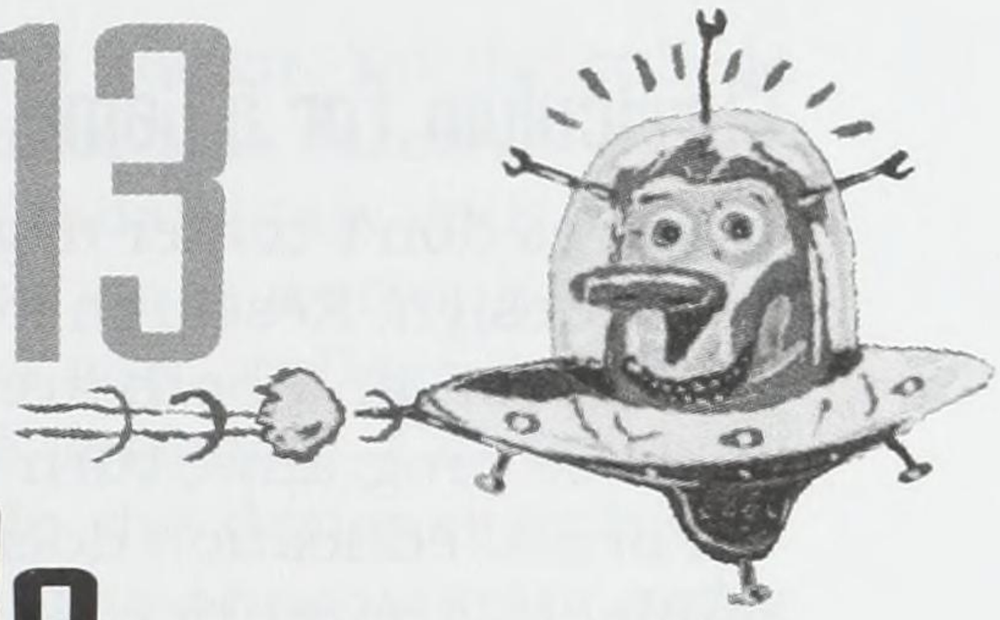
Unlike many adventure games, your career path doesn't have one route to victory. There's no inevitable progression, each step clearly marked, from gamer to gamemaker. If the conversations with the designers in Part Four prove anything, it's that everyone hacks their own road through the gaming jungle.

The uncertainty itself daunts many. They crave the security of an unambiguous path. If this describes you, other careers in interactive electronic entertainment may be more appealing than the life of an independent game designer, at least to start. The quality assurance, product, and public relations departments are excellent launchpads for an electronic gaming career. After a few promotions, designers may be coming to you as a producer or product manager. Another possibility is journalism. A scan of current magazines and books about electronic gaming indicates that there is bountiful opportunity for people who can write insightfully about games.

Any and all of these jobs can eventually lead you back to game design. If there is any one rule about getting a job as a game designer, it's that there are no rules. The next few chapters offer advice and suggestions—not step-by-step guidelines—for folks who don't mind a few bumps and anxious moments en route to fame and fortune in electronic game design. Well, not *that* much fame, and a fortune is what the publisher promises for the game *after* this one, but certainly there will be enough of both to make for a pleasant and satisfying life.

Perfect plans to scale the heights are scarce. A creative mind and lots of energy are the main requirements, but a little luck and a lovable personality won't hurt your chances. With enough of each, you should get a foot in the door and a chance to show someone what you've got.

Chapter 13



Preparing to Be a Game Designer

The uncertainty of professional electronic game design isn't all bad. It's unsettled condition has a positive aspect, too. Since there is no well-worn career path, publishers and developers have few burdensome preconceptions about the appropriate pedigree for someone who thinks up games. Everybody knows that designers can come from anywhere, any walk of life, and so they *do* come from anywhere and any walk of life.

In no other art does the patron or customer invest so much in someone who has not proven his or her skill. A painter may get a commission for a portrait that doesn't exist at the time of payment, but the painter will have certainly painted *something* before acquiring a paying client.

Not so the game designer. That wasn't true when almost all designers were programmers, but that's the situation now. During the era when programmers also did the design work, they got hired by showing a sample of their work. Usually, they hacked an arcade-to-home translation or a game of their own invention. The publisher's techies studied the code, ran the program, and rendered their verdict on the job candidate.

Traditions die hard, even in a future-facing business like interactive electronics. A small work sample is still about the easiest way to get hired. But the current situation is more hopeful for the majority. Even if you can't write a meaningful line of code (one of the author's dubious distinctions), you can still be in the running if you are properly prepared.

Curriculum for a Game Designer

Colleges don't confer degrees, or even offer classes, in electronic game design. Research for this book turned up only two programs of interest, and both are in Canada. For a complete description of these programs, turn to Appendix C, "School for Designers."

Formal education doesn't yet recognize game design as a valid subject. Apprentice designers can attend workshops at the annual Game Developers Conference, but these are of limited help, like most writers workshops. In 1995, though, even the correspondence schools that advertise computer courses on TV didn't have anything for those who wanted to make games and entertainments.

Periodically, a news squib mentions a college course called something like "Video Game Appreciation." It's a good story, when embellished with tongue-clucking about the frivolousness of modern college students, but it's a red herring. These one-shot, oddball courses have no deep or continuing commitment to conveying real knowledge about the product design specialty. They're usually snooze hours for less-than-serious students. They don't teach anything that can't be learned by spending the same hour at the local arcade or in front of a game console.

Just because there aren't specific game design courses doesn't mean that the education system holds nothing for the prospective game designer. There are a lot of opportunities in high school and college to learn things that could help the designer.

Teenagers always want to know if designers need college. "Need" is the pivotal word. There's nothing about four years of university study that teaches anyone to design an electronic game. For every game designer with a PhD there are five who left school after collecting a high school diploma.

Unlike top-level programmers, designers function best if they have a lot of general knowledge. The programmer devotes years of intense practice and experimentation to the mastery of computer coding, which drastically reduces available time for everything else. The designer needs no comparable technical skill, but they have to have a large reservoir of knowledge from which to fashion their interactive entertainments.

Colleges will catch up eventually and add game design courses,

but right now they won't teach anyone to design. Yet the college experience makes the graduate more fit for this career.

The best designers are well-rounded individuals with a great breadth of interests. A liberal arts education is a strong foundation for such a person. Looked at that way, college is a positive step for *any* writer, including game designers. Courses in a number of different disciplines can help the designer in his or her work. Here are some examples, in the approximate order of importance.

English

Written communication looms so large in game design that anything that improves communication skills is worthwhile. Courses in advertisement copywriting and journalism may be particularly useful in learning to write effective proposals.

Computer Science

Designers no longer have to be able to program, but increasing computer knowledge is never a waste of time for a designer. At the least, an introductory course acquaints the designer with the nomenclature and theory of the technology that underlies all interactive electronic entertainment.

Math

Programmers benefit from advanced mathematical study, but the designer's needs in this area are less lofty. Some study of probability theory and statistics shows the designer how to manipulate probability charts. A college-level math survey course enhances understanding of the mathematics of programming. Often, the designer must compile alternatives with varying odds into a chart that the programmer can use as the basis for a chunk of programming. Even those who plan to specialize in action titles need some chart-making know-how.

Logic

A design document is a series of logical statements about the game. A course in philosophy, logic, or symbolic logic hones the author's ability to think and write things logically, which makes it easier for the programmer to transform the plain English document into lines of code.

Draftsmanship

The rudiments of mechanical drawing come in handy for preparing playfield thumbnails and other visual aids that accompany written design documents. Just learning how to read and compile a flow chart can justify a one-semester course.

Library Skills

No one knows everything, but smart people know how to find the missing data. Familiarity with library research methods is going to become progressively more important, since publishers often enrich multimedia titles with ancillary information that relates to the game's theme.

Marketing

The pragmatism of the software industry's marketing professionals may shock the typically idealistic game writer, but effective marketing is an undeniable contributor to the success of commercially distributed electronic games and entertainment products.

Computer Graphics

Some advocate that would-be designers take a computer science course. It may help, but an introductory computer graphics course might prove even more valuable. A better understanding of the visual component of interactive electronic entertainment may allow the designer to contribute more effectively to that portion of the project.

Independent Study

Personal initiative counts heavily in the education of a future game designer. Since there are few teachers, students must double as their own professors.

Books, such as *Game Over* by David Sheff and *The Fall and Rise of Video Games* by Leonard Hermann, furnish inside looks at the industry. They give some advance warning of what to expect in the interactive entertainment world.

The field's interactive entertainment magazines are a good source of information. Alas, the quality of the journalism is spotty, and some writers' expertise is also open to question. As editor and assistant publisher of *Electronic Games*, my loyalties are

obvious, but *EG doesn't* have a monopoly on solid material. The magazines that are filled with colorful strategy maps and one-sentence blurbs are much less useful than those that carry long reviews, interviews, and news analysis.

Out-of-field magazines seldom portray the industry or its products very accurately. *Wired* has many articles about cutting-edge technology, and its investigative reports are also very illuminating. Many of the industry's technologically savvy people read *Morph's Outpost*, so it belongs on the reading list, too.

The Fandom Connection

Electronic Gaming Fandom, about six years old at the present time, is a participatory discussion forum. It's an unorganized network of people who are so interested in games that they go beyond merely playing them.

The main activity is writing, drawing, and editing small-circulation amateur publications called fanzines. Such little magazines have existed for well over a hundred years, but it is only recently that electronic gamers got into the act. There are approximately fifty video gaming fanzines published per month in the United States, and the number of active titles is in excess of two hundred!

The average fan is a male teenager, but it is in no sense restricted only to youngsters. Some of the material is, admittedly, blithering, but it is remarkable how much intelligent commentary and thought-provoking opinion are found in the cheaply copied pages of current 'zines.

Fans get more than practice communicating about games. Through fandom, they can make valuable contacts, among fans as well as pros, which help them turn pro. A designer wannabe who meets a programmer and an artist with complementary ambitions is well on the way to a job-getting product demo disk.

Fanzines, as leisure-time efforts, are prone to dramatic fluctuations in frequency and format. Any addresses and prices listed here would be well out of date within six months. For the latest word, consult the fan columns in *EGM2* and *Video Games*.

Networking the Network

Electronic Gaming Fandom represents a network of about three hundred active game-lovers, plus a few hundred fringe fans. It makes up in single-minded purpose and enthusiasm what it lacks in population.

Local bulletin board services (BBS), national networks, and the Internet constitute a much larger arena. Telecommunications via the computer touches millions of people. Not all, or even most, of them are crazy about interactive entertainment. The cyberverse is so vast, however, that it has spawned hundreds of information sources, including live forums, message boards, pages on the World Wide Web, and real-time chat areas for electronic gaming enthusiasts.

Now I'd like to take this opportunity to preach a little bit: prospective designers who surf the net should avoid involvement with software piracy. Though not as rampant as it was in the mid-1980s, there are still plenty of people trafficking in illegally copied electronic games. Leaving all the other excellent arguments for future denunciations of those sanctimonious thieves, software piracy comes down to "do unto others" for anyone who wants to become a full-time game designer.

Independent designers make most of their money from royalties. The royalty structure makes the creative team dependent on selling enough copies to balance the original advance and generate bonus payments.

Many netters rationalize that they hurt no one when they give away pirated software. Actually, every unauthorized disk takes money out of the designer's pocket.

A game designer is in a position of trust with a publisher or development house. They have access to commercially sensitive information and programs. Anyone with extensive connections with pirates will always be suspect.

Learning the Games

The most important part of a prospective game designer's preparation takes place outside the classroom. A vintner must know wine, an attorney needs familiarity with legal precedents, and a

game designer must gain a thorough knowledge of interactive electronic entertainment.

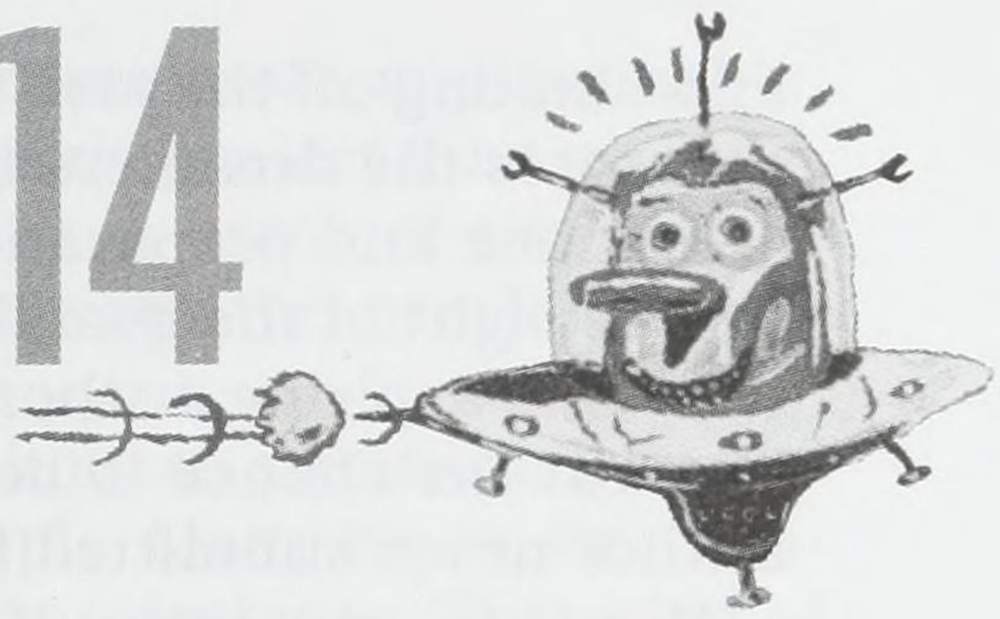
Prospective designers shouldn't worry about how well they play, just that they play video and computer games regularly and frequently. Expert gamers will have an advantage because they get further into a game, and that lets them see elements that a poor scorer can't reach. But this problem is easily cured by watching other, more accomplished gamers whip through the games.

There's no advantage to putting a thousand hours into becoming king-of-the-hill of a fighting game. It's nice to enter initials on the coin-op vanity board or crush friends at the cartridge console, but exposure to a great number of different games is much more helpful.

It's important to do more than just play the games; they have to be dissected. A designer must think like a critic. That means mentally deconstructing a game to examine its parts and see what makes it tick.

The goal of all this analysis is to develop an instinct for what works and what doesn't in a game design. It is especially important to pick apart anything that seems unusually good or bad. Publishers and development houses, the two most prolific employers of designers, are always impressed by an encyclopedic knowledge of electronic gaming.

Chapter 14



Breaking into Game Design

This is where the tourists go home. This chapter is not about game design. It speaks to the question of how to actually become a designer.

Now history, theory, and preparation turn into something tangible. While producing an electronic game may occasionally come across like outtakes from Mel Brooks' *The Producers*, that's the real-world environment in which designers strive for artistic fulfillment and material security.

Until the prospective designer survives this rite of passage, it's all just recreation. Many people enjoy designing and programming games as a hobby, but this book is primarily aimed at those who want to do it full time for the mass audience. As artists, game designers fall into the same category as orchestra conductors and ballerinas: They can't manifest their art alone. The design document only becomes a game with the help and cooperation of other artists and technicians. So until you get hired, your fabulous game will exist as only so many sheafs of paper.

Are You Ready?

Fans of Frank Herbert's science fiction epic *Dune* may remember the Bene Gesserit saying, "Fear is the mind killer." It's applicability to the situation of a budding designer contemplating finding that first job is undeniable. Overcoming the fear of failure,

and shaking off the artistic paralysis that it may induce, is what separates the dreamers from the doers.

Practice and preparation are important, but it's also vital not to lose sight of the goal: becoming a professional designer. More than one aspiring author with folders full of half-finished novels has lost the chance to become a professional writer because the author never submitted his or her work.

How can you know if you're ready to make your move? It's impossible to be certain. No one is ever completely ready, so at some point the candidate must suck it up and plunge ahead. Those who have acquired a comprehensive knowledge of interactive electronic entertainment, formulated a pool of game-concepts, and written up some of these ideas as proposals are primed for the job hunt.

The First Design Job

The apprentice designer can target three types of prospective employers: software publishers, independent development houses, and entrepreneurial start-ups. Each has good and bad aspects.

Appendix B lists all the major publishers of interactive electronic entertainment along with contact names. Since new companies come into the field all the time, and executives jump around a lot, a phone call to the intended recipient will avoid misaddressing materials.

Catching the eye of a software publisher with a mailed-in proposal and landing a contract is the brass ring. It confers instant status, but it is probably the least likely entry path.

Publishers see many, many proposals, most from designers with solid credits. A neophyte will find it hard to stand out in a crowd. The big game companies reject almost all freelance, unagented submissions. There's a chance, though, which keeps independent designers pounding away at the keyboard and publishing executives sifting through stacks of unsolicited submissions.

There may also be a credibility problem. Publishers prefer to play it safe. If the proposal doesn't come from someone who is known to them in some way, even if not as a designer per se, it won't get much attention. Few big operations want to risk a multimillion-dollar project on someone who might not be able to carry through on the job.

Development houses, being smaller, are more approachable. No one is more anxious to find a viable new project than an independent with a heavy payroll to support. A batch of sharply defined proposals can lead to a meeting with the president or a product manager.

It also helps that most development houses are started, and to some extent controlled, by product people. Many times, its president was an enterprising programmer or designer. That kind of built-in sympathy and understanding can't be bought.

Even if the independent developer doesn't snap up one of the concepts, such dialogs often lead to work. Many designers have started by working on modules or otherwise assisting a project's chief designer. It's still not doing your very own game, but it's definitely a foot in the door—and it's more likely to occur than walking in off the street and selling a new game design.

Or, take your game concept, gather some other hungry programmers and artists around you, produce the game yourself, and then shop it around. Start-up ventures are characterized by small front money, high risk, and great financial reward and prestige down the line if the game hits. Those with an entrepreneurial bent who don't mind taking chances may find this the most satisfying strategy.

id Software is a model for such a launch. Diverse creative talents came together to work on a game in which they all had faith, *Wolfenstein 3D*. When the crew had something playable, they issued an inexpensive, partial version through Apogee Software, a company that specializes in shareware titles. *Castle Wolfenstein 3D* became a sensation. That allowed id to do a full-dress commercial edition, *Spear of Destiny* (FormGen). From these twin successes has come lucrative licensing deals for the game-engine, a mega-hit called *Doom II*, and matching Ferraris for John Carmack and John Romero.

They had the goods, and they made it pay off handsomely. Most start-ups don't end so happily. The designer must face the possibility that no product will emerge from months of effort, and that the enterprise will flop. But even a stillborn project may not be a total waste. Even if it goes begging among publishers, it may serve as an acceptable work sample when submitted with other, unrelated proposals.

Finding people to join a start-up isn't as easy as locating a publisher or an established development house. Among the best places to look are fanzines, user groups, and among low-echelon creative people at large companies. A designer who can rally skilled people around a concept can cut through the red tape and become a practicing designer in less than a year.

Submitting Your Brainchild

The following guidelines for sending proposals to publishers aim to produce the best possible impression at the receiving end. They're mostly a matter of common sense, but they're worth reviewing here. Deviations from the norm aren't necessarily lethal, but they are unlikely to advance the designer's cause.

The basic design proposal package has three parts: the document, the cover letter, and the proposal. The former sets the stage for consideration of the latter, so a little extra effort won't be misplaced.

The proposal sells the game, but the cover letter markets the designer. It is important to acquaint the publisher with your background, especially things that suggest you would be good game designer material. Enthusiasm and salesmanship play a role in the cover letter, but a little warmth and humility aren't bad counterpoints. The recipient already *knows* you think your game is wonderful. Better, and more professional, to let the proposal speak for itself.

Keep the cover letter short. One page is optimal, and more than two should set off alarms. The worst thing a submission letter can do is exhaust the reader's patience before he or she delves into the actual proposal.

If the designer has an extensive professional gaming history, the best course may be to take all that information out of the cover letter and include a résumé. Restrict it to a single page by leaving out extraneous information about nongaming jobs that don't apply to the project.

The design should be laser printed on good white paper. Anything less may put doubts about commitment into the evaluator's mind. (Yes, bubblejet and inkjet printers are acceptable, but stay away from dot matrix on tractor-fed paper.)

Don't forget to include your address and phone number for return contact. Any e-mail address and fax number should also be prominently mentioned. A self-addressed, stamped envelope insures that the manuscript will come back if the publisher doesn't accept it.

The Indirect Approach

Many designers travel a more circuitous path. Only a few people jump right into design; others edge into it from other electronic gaming jobs.

The likelihood of success is much greater this way than with a frontal assault. Industry visibility yields contacts in the right places, and it's an advantage to have someone at a publisher who will view a proposal with a friendly eye. Knowing the right people won't sell a design, but it removes obstacles and mitigates the curse of being a first-timer.

Working in allied specialties builds an employment record that reassures deal-makers. A resumé with deep experience won't sell a project, but it definitely earns a hearing. And again, even if the specific idea doesn't sell, publishers and development houses treat industry colleagues more respectfully than they do unknowns.

There are several jobs that may lead to designing opportunities: journalism, quality assurance, and documentation.

Journalism

Writing about electronic games is an excellent precursor to creating them. It's a good way to get paid for acquiring product knowledge, and nothing beats high-volume writing for improving communication style and effectiveness.

Good journalists also have surprising *entrée* to the highest echelons of electronic publishing. Journalists get many chances to display their gaming savvy, and an impressed executive can be a conduit to a design job.

The author is an applicable case history. Brian Fargo of Interplay Productions said something like, "You write so well about games that I'll bet you could design one." Brian had a contract for a hardboiled detective adventure for Activision, which part-

ner Bill Kunkel and I turned into *Borrowed Time,* later republished by Virgin Games as *Time to Die.*

Books won't establish an industry-wide presence as rapidly as magazines, but writing strategy guides has been a bridge for more than one would-be game creator. The give-and-take between a book author and the design/development team sparks friendships that can become working relationships.

Quality Assurance

Publishers hire small armies, both in-house and in the form of independent providers, to meticulously examine games for technical flaws and other mistakes. These jobs are relatively easy to get, compared to a position as a designer, and the sky's the limit once inside the company.

This alternative is open primarily to would-be designers who are also very strong players. A Q/A person must learn a game quickly and play it thoroughly to compile the report, so those who can't get past level two in anything need to seek elsewhere.

Documentation

Rulebooks don't pay much, and doc authors don't usually share in the royalty pool. That doesn't matter. Writing the rules is an affiliation with a game project, the necessary crack in the wall.

Some doc writers become so proficient that producers start asking for them on projects. Sooner or later, a game deal will come along that has a niche for a newcomer.

Whatever the route, the desired result is the same: designing a commercially published game. Turning professional seldom goes smoothly, but with perseverance and talent, it *will* happen.

Chapter 15

Designing the Future

If reading about the history of electronic games in Part One teaches anything, it's that interactive electronic entertainment never stands still. Fueled by fast-track computer technology, the game business goes through revolutionary upheavals every few years. The average platform has about a five-year prime, and a game sells 90 percent or more of its total in the first twelve months.

Veteran designers who can't adapt fall by the wayside. Newcomers who are unprepared simply don't get hired. Just like in a first-person, flying and shooting game, anticipating what's coming is the first step to survival. Anyone with long-range career plans must think about tomorrow as well as today.

What will game design be like in, say, the year 2000? The specifics are hard to pin down for next year, let alone next century, but some things appear fairly certain. No one can say what kind of game will be the *Doom* or *Street Fighter II* five years from now, but that's not necessary. The important thing is to estimate the future role of the game designer.

The Twenty-First-Century Gamer

Ask the average person to describe the people who play electronic games, and they'll mumble something about how it's mostly kids. That's the image the media, including game company commercials, pound into our brains every day, so that's the

stereotype someone is likely to spout when asked for an off-the-cuff comment.

Likely, that is, unless the question is asked of one of the millions of adults who play. The stereotypical image of the teenage video wizard is outmoded in 1995, and it will become increasingly ludicrous over the next few years. Interactive electronic entertainment has always appealed to all ages, and new delivery modes will make these games more available to the whole family.

The audience is becoming more diverse in demographics, interests, and gaming experience. As hardware capabilities soar, games will reach out to people who weren't captivated by the more primitive products of prior software generations.

TV-quality visuals and full stereo sound will bring millions of people into the gaming hobby, and the activity will continue to grow in popularity. Lower hardware and software prices will help, too. Some people will always prefer more passive entertainment, but the audience for interactive fun is far from topping out.

Marketing to the teen hard core, a favorite strategy today, won't work as well in five years. Titles that isolate this traditional video game segment will chalk up decent sales, but they won't capture the full mass market. The predicted average gamer is likely to be an adult, average age twenty-five or so, and many more women will be participating in the hobby, too.

Look for the return of niche marketing. Popular in the computer entertainment field in the 1980s, it fell out of favor in the mass market-oriented 1990s. Just as narrowcasting is replacing broadcasting in video programming, aiming products directly at a highly committed sliver of the market will be a viable option.

The youth mass market will be more like a hunk than a sliver. It will be the biggest niche in 2000, as it is in 1995. Colorful, high-speed, physically demanding games will still have millions of fans, but titles with even wider appeal could sell two or three times as much as contemporary cartridge hits.

The Hardware Horoscope

If the direction of mass market technology could be forecast accurately, the publisher could retitle this book, *How to Make a Billion Dollars in the Stock Market.* A volume that correctly picked

the top companies five years down the line would be an invaluable investment guide.

The fates of individual hardware and software companies are an unknown quantity. Trends are another matter. And it is the industry's broad sweep that will define twenty-first-century designers, not the ups and downs of the various corporate players. Anyone can get stuck in the wrong place. If the designer plans properly, it is possible to rebound from even such a setback.

Telecommunications will be at the center of next-century gaming. Stand-alone systems won't disappear, but millions of Americans probably will get most of their games over the wire. Retailers who specialize in cartridge rental will be hunting for other uses for their shelf space. Brand new titles may still be a hot rental item, but download-on-demand will cut into rental revenues.

So much is said and written about the information highway that it is easy to slide into skepticism. It couldn't possibly be as important as the experts say. But it can be. In fact, it is. By the end of the decade, a sizable number of American homes will be connected to the highway through computers, cable, or telephone.

The highway won't keep people from acquiring several different machines to feed their gaming habit. An *Electronic Games* poll determined that its readers currently own almost three systems each. Multiplatform ownership will continue to be strong, though all that clutter will encourage development of a single hardware shell into which users can load a variety of operating systems, and the software for each. In fact, access to software may prompt consumers to buy into more platforms.

The machines' capabilities will be significantly greater than in 1995. More RAM, faster data access, and higher operating speeds are guaranteed. Manufacturers already have these improvements in development, and they will be everyday stuff by 2000.

Larger capacity storage media is also a certainty. An alliance of international hardware giants has agreed to use the DVD (digital video disk) format, and even better data storage media may be in the offing.

Hardware that presents true three-dimensional images is an inevitability. Companies have pumped millions into its development because it's perceived as a major consumer desire. No one can say today how important 3-D will be when it arrives, or how quickly it will penetrate the mass market.

Stand-alone virtual reality units are much easier to market because they don't require families to buy a new TV set. Despite the equivocal quality of virtual reality helmets to date, manufacturers are making tremendous progress toward perfecting the technology. VR will be available in 2000, but the price of a good system may still be high for the mass market.

Software 2000

The computer and video game segments will blur together, submerged by the rising tide of multimedia. Traditional floppy disk and cartridge games will be a minor part of the business, and few sizable projects will focus on anything except games that incorporate the latest multimedia technology, both on CD-ROM and over the wire.

When grading games, some critics penalize current multimedia entertainment products for their relatively low level of interactivity. The noninteractive FMV (full-motion video) cut scenes look great, but they don't provide any game-play. That won't be acceptable game development in the twenty-first century, however. The next decade's multimedia software will benefit enormously from the anticipated hardware enhancements. Speedier data streaming will make it possible to use more interactive ideas, creating a class of games that may well resemble today's movies in audiovisual quality.

Although technology will extinguish the difference between CD-ROM disk and cartridge games, there will still be two basic types of interactive electronic entertainment. The key determinant won't be whether the program is on mylar or silicon, but it will be the conditions under which it is played.

There are two main environments: the computer workstation and the living room media video center. The former emphasizes extended play by individuals against the computer, while the latter intrinsically leans toward streamlined, quick-start, multiplayer entertainment with plenty of eye- and ear-candy for those who await their turn.

Access to opponents via the modem will soften this distinction, but not obliterate it. More people will get into two- and multiplayer games through computer networks, and these will definitely include lots of real-time, head-to-head competition.

Publishers will no longer picture compu-gamers as lonely individuals locked away from the world, but rather as members of the digital community.

Today's *Doom* phenomenon signals the true dawn of the age of telegaming. Many computerists who played solitaire and complained about the predictability and coldness of games directed with artificial intelligent adversaries are finding the human factor an exhilerating enhancement to the play experience. Games without modem and network support will be rarities in 2000, and many titles will be obtainable only over a wire.

Game Designers of Tomorrow

Projects will continue to get larger through the millennium. Consumer expectations about the depth and size of programs head upward relentlessly, and there will be no slowdown. It's human nature to want the new games to be better and bigger than last year's, and publishers must outdo themselves or fail to generate the necessary retail excitement for their releases.

Giving buyers more of everything means that a larger number of creative people will have to work longer hours to finish a top-shelf product. Some publishers may cut down the number of active projects and focus resources on a smaller number of titles. This strategy holds down total sales as well as expenses, so top management and profit-hungry stockholders won't be very happy with it.

That's why most operations will spend money to beef up the design/development team. It makes more sense to increase personnel rather than extending the development cycle because any product that requires more than fifteen months to produce risks being bypassed by technology. If anything, publishers will try to cut development time to a year or less to overcome the negative consequences of excessive time-lag.

Expanding the design and implementation crew means that fewer authors will get assignments to design a game from boot to shutdown. Top designers will find themselves running teams of assistants. That's a bonus for anyone contemplating a game design career because those larger crews are sure to have slots for assistants that will be ideal for the newcomer.

If this happens, the best route to a design job will be to join a publisher or development house as an assistant instead of sub-

mitting a proposal. The projects will start with time-proven chief designers, but it will also be less risky to put an untried game writer onto the payroll, since the whole project won't die if he or she can't cut it.

Larger projects, and crews to execute them, will put greater emphasis on an exhaustive design specifications document. If several designers share responsibility for one project, then it will be harder for implementation specialists. A comprehensive, and rigorously checked, specs document will be needed to keep confusion to a minimum with so many people working on aspects of the same project.

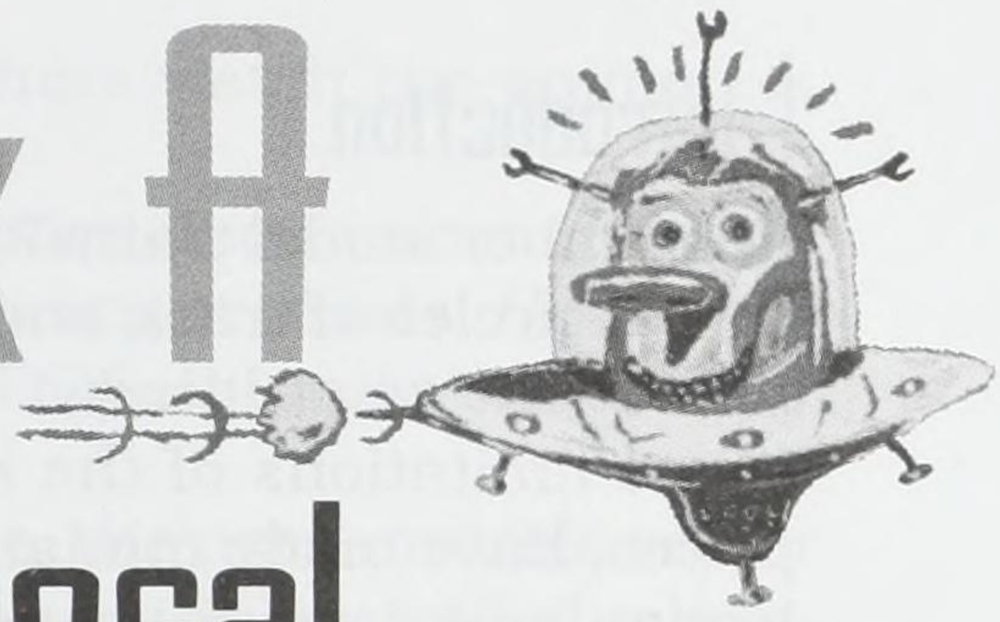
The nature of the designer's contribution is in flux, and it could be quite different by the dawn of the new decade. Most designers now work indirectly. They write the specifications document, hand it to the producer, and then cheer from the sidelines while the implementation crew grapples with the problems of turning creative dreams into data streams.

In 2000, the input may be much more direct in a majority, or at least a sizable minority, of projects. Implementation tools are increasingly user-friendly. Many don't require programming knowledge, just a short course in how to use them. These tools allow the designer to put an approximation of the design right onto the screen without an intermediary. A design will still need a lot of work after that, but it gives the designer a closer connection to the finished product than a stack of written instructions.

The vogue for game engines will also help designers get more involved in actual implementation. Game engines aren't playable products, but they can be used to create a whole series of products that share a common structure but differ in graphics, sound, and storyline.

Game engines are like an empty bottle into which the content of a new game is poured. Designers can work out a game in rough using an engine because engines frequently require no programming expertise and they allow for major creative input. Of course, programmers will still be necessary to both prepare those engines and to do the finishing touches and nonstandard aspects of a project. However, the ability to put even a rough approximation of what is desired onto the screen will insure that designers remain the principal providers of game content in the new century.

Appendix A



A Sample Proposal

Nothing beats a good, practical example. Chapter Ten in Part Three provides a template for an entertainment prod uct proposal. In this appendix, I've used this template for the fictional proposal for *Highballin': Electronic Model Railroading*. In this way you can see how the general principles apply to a specific game.

KATZ KUNKEL WORLEY
Las Vegas, NV
Telephone: 702-555-5555 Fax: 702-555-5555

Highballin': Electronic Model Railroading

Design Proposal Submitted By

Subway Software
(Arnie Katz, Bill Kunkel, and Joyce Worley)

June 15, 1995
CONFIDENTIAL
Copyright 1995, Katz Kunkel Worley, Inc.

I. Introduction

Remember model trains? Just about every adult has fond memories of circles of track and engines that huff and puff. Sad to say, that's just what it must remain for most of us, a memory. The space limitations of the modern apartment, coupled with steep prices, have made miniature trains little more than a fond recollection and a wistful yearning.

Fewer and fewer families have room for their own model railroads. The trend toward space-saving, smaller gauge systems has reached a practical limit with "N" scale, since it's hard to get personally involved with a locomotive that is smaller than a cigarette lighter.

Despite these obstacles, interest in model railroading is growing in the United States at the present time. Ironically, interest has focused on large scales rather than the small gauges. Most people can't afford to buy much of this stuff, despite their great interest in it. A simple set of *Highballin'* O-gauge trains costs at least $300! The average set is closer to $600. That doesn't count a table, a better power supply, or track turnouts (what non-railfans call "switches") to allow the train to do more than go round and round endlessly. One O-gauge automatic track switch (and it takes two to create an alternate route) costs a lot more than this product.

Highballin': Electronic Model Railroading puts a full, working tabletop railroad within reach of everyone's pocketbook. People can enjoy the challenge and satisfaction of running a rail empire without worrying about where to put the equipment—or how to pay for it. In other words, this multimedia product allows people to enjoy a hobby at a level that would be impossible without this technology.

II. Product Overview

Highballin': Electronic Model Railroading is exactly what the name implies, a detailed simulation of O-gauge model trains. Its "construction kit" format lets the user create a completely scenic layout.

Highballin' is essentially a one-player activity game, much like the hobby on which it is based. Also like actual model trains, *Highballin'* is also appealing for small groups. Participants take

turns running the trains while the others watch the animated, real-time model railroad.

The primary platform for this interactive electronic model railroad is computer multimedia. In light of current trends, *Highballin'* should run under Windows as well as DOS. It is also suitable for 3DO, PlayStation, and Saturn.

With the program in operations mode, the user controls one or more model railroad trains. The trains can be controlled manually or run on automatic. The trains can pick up and deliver cargo and passengers at animated loading/unloading sites in response to orders from customers, or they can just run through town and country.

A strong emphasis on audiovisual effects conveys the "feel" of running an accessory-laden *Highballin'* model railroad. This includes authentic train sounds, smoking locomotives, animated loading and unloading, functional track signals, and much more.

Each user can personally design a dream layout. With this program, anyone can construct the kind of sprawling panoramas previously found only at model railroad clubs and the homes of the rich and famous. It is simple to design a realistic operation or one with the toylike ambiance of an old O-gauge set.

Highballin': Electronic Model Railroading uses the audiovisual capabilities of CD to portray train layouts with all the fun and play-possibilities of the metal-and-plastic variety.

III. Opening Sequence

When the disk boots, the user sees an antique-looking model train catalog. It opens to a double-page spread of a gadget-laden model train layout.

As train sounds begin, softly and then rising, the image zooms into the catalog, inside the frame of the layout picture. It focuses on a train building up steam at a rural station. The picture morphs from a two-dimensional image to a pseudo-three-dimensional one.

The engine sounds are loaded as the train starts to pull away from the station. The scene rotates so that the train is coming directly out of the screen at the player. As it roars into the foreground, the title comes up superimposed: *Highballin': Electronic Model Railroading.*

IV. Graphics

Multimedia graphics make the difference in *Highballin': Electronic Model Railroading.* Its detail and animation make actual model train operating cars and accessories look clunky by comparison!

Perspective: A variety of visual presentations enliven the simulation. The construction kit employs an overhead view, as does the control panel. During ordinary running, the hobbyist sees the train from a "three-quarter" angled side perspective. A first-person view is available as an option.

Animated Sequences: When a player wants to drop off or pick up cargo at a siding, a close-up in modified side perspective replaces the usual one. This allows the program to present a little scene, complete with colorful workers and plenty of detail, at that site. When the job is done at a site, the regular view of the layout returns.

Although the designer conceives of these scenes as rendered artwork, to better harmonize with the texture-mapped polygon scenery, they could also be presented in full-motion video. The expense and difficulty of shooting the loading and unloading scenes and a desire to emulate the feel of a model train set are factors favoring use of artwork.

V. Sound

Highballin': Electronic Model Railroading will sound as well as look like a real train set. Digitized sound effects, synchronized with on-screen events, enhance the impression of solid objects in motion.

Digitized speech also plays a role. During operation, the user hears the voices of other workers, passengers, station personnel, and load bay supervisors.

Music is available as an option. The gamer selects songs from several different eras.

VI. Play-Action: Running the Railroad

Since this is a simulation of model railroading, the program provides several different ways to enjoy the layout.

Round and Round

The model railroader can put operations on "automatic" and let the trains run around a small-scale countryside. It makes a nice demonstration for friends or a fascinating (and relaxing) solo experience.

Assembling Trains

Because it is possible to build large railroad yards with *Highballin': Electronic Model Railroading,* a prime activity is assembling and breaking down trains. The complex switching situations are like chess problems, logical puzzles that stimulate the gray matter.

Single Train

The user concentrates on the journey of a single train from roundhouse to the final destination.

Rail Empire

The simulation mode tests the hobbyist's ability to coordinate and control an entire, multitrain rail line. The computer generates pick-up and delivery orders, some of them time-critical, which the user executes in the most efficient manner possible.

This flexibility is one of the main strengths of *Highballin': Electronic Model Railroading.* Some users will focus on the yard, while others will prefer timetable operation.

In the simulation mode, this program bears some resemblance to simulators like *F-15 Strike Eagle* and *Flight Simulator II.* There is no winner or loser in the traditional gaming sense. If the hobbyist keeps moving the goods safely and profitably, he or she is a successful railroader.

VII. Play Mechanics

Highballin' features a variation of the control system actually used by large model railroad clubs. The hobbyist, as chief dispatcher, must make sure that passengers and freight arrive where and when they should on time. At the beginning of each "game/day," the program generates shipping orders and passenger schedules for the ensuing twenty-four hours. The gamer determines which cars will be needed to satisfy customer demand, pulls to-

gether the appropriate trains, and moves the merchandise to the designated sites.

A layout schematic facilitates control of one or more trains at the same time. Signal lights instantly inform the operator which way each switch (turnout) is thrown and the status at every siding. The control panel for various layout functions, activated with an on-screen cursor, will be very similar to ones found on big-time layouts.

VIII. Difficulty, Playability, and Play-life

There are three different ways to adjust the difficulty of the operations mode of *Highballin': Electronic Model Railroading*. These are:

- Raise or lower the number of work orders per day.
- Increase or decrease passenger service to make timed operation more or less stringent.
- Alter the frequency of unusual occurrences. Unplanned factors make it harder to keep the railroad running smoothly.

Highballin' is an open-ended product with a virtually unlimited play-life. The user can construct a variety of layouts, mix rolling stock, relocate depots, and change the scenery to keep the experience fresh and involving.

IX. Special Feature: The Construction Kit

Viewing the layout construction board from overhead perspective, the user employs the control. The controller guides an on-screen pointer that precisely positions each part on the layout as desired. If the image has an associated animation, the program automatically incorporates it into the set-up.

Data bank categories include: locomotives (diesel and steam), rolling stock, terrain/scenery, people, fauna and flora, railroad and private buildings, signals, and all the other things that make model railroading so entertaining.

Selecting an industry automatically implements the animations associated with it. Some users may want to disable such elements, but most will get a kick out of watching the little cows lumber into the waiting box car.

When a train pulls into an industrial siding, a secondary screen appears in the center of the display and shows the workers loading or unloading cargo, which gives *Highballin': Electronic Model Railroading* unique visual appeal. This facet is bound to spark golden memories of the "working cars" that so enlivened Highballin' and American Flyer.

The digital rail baron can select rolling stock (engines and cars) from several historical periods, from the golden age of steam to the most modern diesels and electrics. Each car can be further customized with railroad logos and color schemes.

X. Market Perspective

Model railroading has wide-spectrum appeal to both adults and children. Older hobbyists will recall their childhood train sets with great nostalgia, while kids will see *Highballin': Electronic Model Railroading* as a chance to have the kind of layouts showcased in features in the model railroading magazines.

Even those who own trains will find *Highballin': Electronic Model Railroading* entertaining and appealing. Few can aspire to ownership of the kind of train layouts that can be created, saved, and repeatedly operated with this program.

Several minor attempts to simulate model railroading have failed in the past, mostly because the technology to present the necessary images did not exist at the time. The success of Sid Meier's railroad strategy game indicates that there is widespread interest in the subject matter.

The publisher may want to consider acquiring a license to further strengthen the image that this is real electronic model railroading. If so, our recommendation is Lionel.

XI. Summary

Highballin': Electronic Model Railroading combines infinite play possibilities with potent eye and ear appeal. We are enthusiastic about designing this product and working with whatever implementation specialists the publisher deems appropriate.

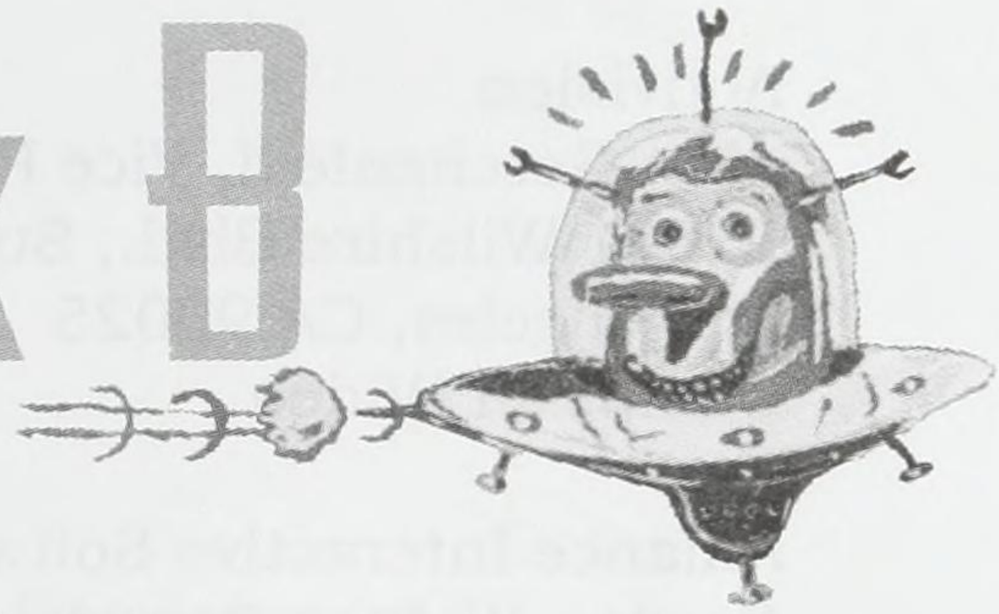
We will produce complete design specifications and make all necessary design revisions as they are ordered. We are experi-

enced in working with both in-house and independent implementation groups.

The three founders of Katz Kunkel Worley, Inc., have worked in the interactive electronic entertainment field since the late 1970s. The company provides diversified services, including consulting, game design, product evaluation, documentation, and journalistic packages to many leading companies. The KKW brochure, which explains the company's activities in complete detail, is available on request.

Copyright 1995, Katz Kunkel Worley, Inc.

Appendix B



Contacts

Those who are serious about careers in game design should focus their efforts on writing proposals and sending them to software publishers. Here are the addresses, contact names, and product focus of a wide range of them.The product focus is listed in descending order of the publisher's specialty. Since companies and executives move around a lot, make sure to call to verify this information before mailing your proposal.

Access Software, Inc.
David Ashby, Vice President of Sales
4910 W. Amelia Earhart Drive
Salt Lake City, UT 84116
Computer/Multimedia

Acclaim Entertainment, Inc.
Paul Samulski, Vice President, Product and Creative Development
71 Audrey Avenue
Oyster Bay, NY 11771
Video Games

Accolade, Inc.
Elsie Derrivas, Administrative Assistant
5300 Stevens Creek Blvd.`
San Jose, CA 95129
Video/Computer

Activision
Alan Gershenfeld, Vice President of Creative Affairs
11601 Wilshire Blvd., Suite 1000
Los Angeles, CA 90025
Computer/Video

Alliance Interactive Software, Inc.
Gordon Walton, President
1859 North Pine Island Road #103
Plantation, FL 33322
Multimedia

American Laser Games
John Pasierb, Vice President of Engineering and Development
4801 Lincoln Road NE
Albuquerque, NM 87109
Computer/Multimedia

American Sammy
Rick Rochetti, Executive Vice President of Sales
901 Cambridge Drive
Elk Grove Village, IL 60007
Video Games

American Technos, Inc., Family Entertainment Division
Greg Rice, Vice President
19200 Stevens Creek Blvd., Suite 120
Cupertino, CA 95014
Video Games

AMTEX Software Corp.
Lisa Moore, Marketing Manager
P.O. Box 572
Belleville, Ontario, Canada K8N 5B2
Computer/Video

Aris Entertainment
Sandra Mueller, Creative Director
310 Washington Blvd., Suite 100
Marina Del Rey, CA 90292
Video/Computer

ASG Technologies, Inc.
Mark Owen, Vice President of Product Development
1601 Civic Center Drive, Suite 203
Santa Clara, CA 95050
Video Games

Atari Corp.
Ron Beltramo, Director of Sales and Marketing
1196 Borregas Avenue
Sunnyvale, CA 94089
Video Games/Multimedia

Atlus Software, Inc., Entertainment Division
Grady Hunt, Product and Marketing Manager
17145 Von Karman Avenue, Suite 110
Irvine, CA 92714
Video Games

Avalon Hill Game Company
Jim Rose, Operations Director
4517 Hartford Road
Baltimore, MD 21214
Military Simulations

Baker & Taylor, Inc., Baker & Taylor Software
Al Strutz, Vice President of Multimedia
237 W. Jefferson
Naperville, IL 60540
Multimedia

BAO Simulation Products
Nick Medici, General Manager
2004 Fox Drive, Suite G
Champaign, IL 61820
Flight Simulators

Broderbund Software, Inc.
Kathleen Burke
500 Redwood Blvd
Novato, CA 94948-6121
Computer

Bullet-Proof Software
John Knoll, Product Development Manager
8337 154th Avenue NE
Redmond, WA 98052
Video Games

Cambrix Publishing
Mark Honorof, President
6269 Variel Avenue, Suite B
Woodland Hills, CA 91367
Multimedia, specializing in databases and edutainment

Capcom U.S.A.
Steve Hanawa, Vice President of Research and Development
or
Richard Moore, Director of Development
475 Oakmead Parkway
Sunnyvale, CA 90486
Video Games

Capstone Software
Linda Zack
7200 NW 19th Street, #500
Miami, FL 33126
Computer

Compton's New Media, Inc.
Steven Marder, Director of Entertainment Development
2320 Camino Vida Roble
Carlsbad, CA 92009
Multimedia

Creative Multimedia Corp.
Len Jordan, Vice President of Marketing
514 NW 13th Avenue, # 400
Portland, OR 97209
Multimedia

Crystal Dynamics
Steve Groll, Submissions
87 Encina Avenue
Palo Alto, CA 94301
Video/Computer

Davidson & Associates, Inc.
Mike Albanese, Director of Third Party Development
19840 Pioneer Avenue
Torrance, CA 90503
Computer Edutainment

Deadly Games/Chaos Tech
Rene Vidmer, President
P.O. Box 676
11 Aelfie's Way
Bridgehampton, NY 11932
Video/Computer

Disney Software
Mike Roggero, Business Development Manager
500 S. Buena Vista Street
Burbank, CA 91521-8462
Edutainment, Computer and Video Games

Dongleware
Catherine Arcolio
35 Howard Street
Cambridge, MA 02139
Video Games

Electro Brain Corp.
Tony Chiodo, Production Manager
573 East 300 South
Salt Lake City, UT 84102
Video Games

Electronic Arts
Send a postcard with your name and address marked
"Attention: product submission request" to
1450 Fashion Island Blvd.
San Mateo, CA 94404-2064
Video/Multimedia

FCI
Interactive Software Department
Makoto Morohoshi, Director
150 East 52nd Street
New York, NY 10022
Video/Multimedia

First Star Software, Inc.
Peter Jablon, Owner
666 Lexington Ave
Mt. Kisco, NY 10549-3632
Video/Computer

Gametek, Inc.
Elizabeth Curren, Vice President of Product Development
2999 N.E. 191st Street, 5th Floor
N. Miami, FL 33180
Video/Multimedia/Computer

Hi Tech Entertainment
Jill Grey, Marketing Manager, Video Games
584 Broadway, Suite 509
New York, NY 10012
Video/Computer/Multimedia

Hudson Soft U.S.A., Inc.
Lou Moiani, Consulting Specialist
400 Oyster Point Blvd., Suite 515
S. San Francisco, CA 94080
Video Games

Humongous Entertainment
Ron Gilbert, Production Manager
13110 NE 177th Place, Box 180
Woodinville, WA 98072
Computer Edutainment

I-Motion, Inc.
Yves Blehaut, Chief Operating Officer
1341 Ocean Avenue, Box 417
Santa Monica, CA 90410
Computer

Impressions Software
Production Manager
222 Third Street, #0234
Cambridge, MA 02142
Computer

Interplay Productions
Business Development
17922 Fitch Avenue
Irvine, CA 92714
Video Games

IVI Publishing
Margaret Vergeyle, Production Manager
7500 Flying Cloud Drive
Eden Prairie, MN 55344-3739
Multimedia

Jasmine Multimedia Publishing
Jay Samit, President
6746 Valjean Avenue
Van Nuys, CA 91406
Multimedia

JVC Musical Industries, Inc.
Yoji Takanaka, Production Manager
3800 Barham Blvd., Suite 305
Los Angeles, CA 90068
Video/Multimedia

Kemco America, Inc.
Oliver Miyashita, Marketing Manager
Westpark H, 8725 154th Avenue NE
Redmond, WA 98052-3878
Video Games

Knowledge Adventure, Inc.
Graeme Weston, Director of Business Development
4502 Dyer Street
La Crescenta, CA 91214
Multimedia
Call 818-542-4200 for submission guidelines
before sending anything by mail.

Koei Corp.
Mark Hughes
1350 Bayshore Hwy., Suite # 540
Burlingame, CA 94010
Video/Computer
with an emphasis on strategy

The Learning Company
Diana Jane Carns, Vice President of Product Marketing
6493 Kaiser Drive
Fremont, CA 94555
Edutainment

LucasArts Entertainment Co.
Jack Sorensen, Business Director
P.O. Box 10307
San Rafael, CA 94912
Multimedia/Computer/Video

Luminaria
Jeff Cretcher, President
152 Brewster Street
San Francisco, CA 94110
Multimedia
Send introductory letters only.
Unsolicited submissions will be returned unopened.

Mandingo Entertainment, Inc.
Peter Van Brunt, President
345 North Maple Drive
Beverly Hills, CA 90210
Video Games

Maxis
Bob Derber, Vice President and General Counsel
2 Theatre Square
Orinda, CA 94563
Computer/Multimedia
with an emphasis on complex simulations

Merit Software, Inc.
Roddy McGuinnis, Director of Product Development
13707 Gamma Road,
Dallas, TX 75244
Multimedia

SAI Incorporated
Scott Ciancio, Producer
201 Bellevue Blvd., V.O.P.
Newark, DE 19702
Computer, with an emphasis on sports sims

MicroProse Software, Inc.
Debbie Wahler, Director of Marketing
180 Lakefront Drive
Hunt Valley, MD 21030
Computer/Video

Microsoft
Anthony Garcia, Business Unit Manager
One Microsoft Way
Redmond, WA 98052-6399
Computer/Multimedia

Mindscape (formerly Software Toolworks)
Jim Fisher, Senior Vice President of Entertainment
60 Leveroni Ct.
Novato, CA 94949
Video/Computer/Multimedia

New World Computing, Inc.
Mark Caldwell, Vice President of Product Development
29800 Agoura Road #200
Agoura Hills, CA 91301
Video/Multimedia

Nintendo Of America
Don James, Director of Product Development
4820 150th Avenue N.E,
Redmond, WA 98052
Video Games

Ocean of America
Ray Musci, President
1855 O'Toole Avenue, Suite D-102
San Jose, CA 95131
Video Games

Playmates Toys, Inc.
Diane Teigiser, Director of Licensing and Merchandising
16200 Trojan Way
La Mirada, CA 90638
Video Games

Psygnosis Limited
Cassandra Beyer, Assistant to the President
675 Massachusetts Avenue
Cambridge, MA 02139
Computer/Video/Multimedia

Quantum Quality Productions
Bruce Williams, CEO
495 Highway 202
Flemington, NJ 08822
Computer/Multimedia

Sanctuary Woods
Mike Scott, Vice President of New Business Development
1875 Grant Street
San Mateo, CA 94402
Computer/Multimedia/Video

Sega of America, Inc.
255 Shoreline Dr., #200
Redwood City, CA 94065
Video/Multimedia
Call 415-802-3848 for product submission info-line.

Sir-tech Software
Linda Sirotek, Product Planning and Acquisition
PO Box 245
Ogdensburg, NY 13669
Computer/Multimedia

Software Sorcery
Daniel Cota, Vice President of Sales
5405 Morehouse Drive
San Diego, CA 92121
Computer/Video

Software Toolworks—see Mindscape

Square Soft, Inc.
Doug Smith, Vice President of Product Development
8351 154th Avenue, NE
Redmond, WA 98052-3865
Video Games

Strategic Simulations, Inc.
Lee Crawford, Technical Assistant One
675 Almanor Avenue, Suite 201
Sunnyvale, CA 94086
Computer/Video
with an emphasis on role-playing and simulation games

Suncom Technologies
Dan Zemaitis, Marketing Manager
6400 W. Gross Point Road
Niles, IL 60714-4508
Computer/Video

Sunsoft
Sun Corporation of America
David Siller, Vice President of Product Development
11165 Knott Avenue
Cypress, CA 90630
Video Games

Taito America Corp.
Larry Stalmah, Vice President of Production
390 Holbrook Drive
Wheeling, IL 60090
Video Games

Takara U.S.A. Corp.
Kinya Tago
230 Fifth Avenue, Suite 1201-6
New York, NY 10001
Video Games

Take 2 Interactive Software
FJ Lennon, Vice President of Creative Development
1004 Ligonier Street, 3rd Floor
Latrobe, PA 15690-1837
Video/Multimedia

Tecmo, Inc.
Emily Nelson, Marketing Manager
Sequoia Commerce Center
19260 South Van Ness Avenue
Torrance, CA 90501
Coin-op/Video

Tetragon, Inc.
Mark Skaggs, President
269 W. Renner Pkwy., #200
Richardson, TX 75080-9604
Video Games

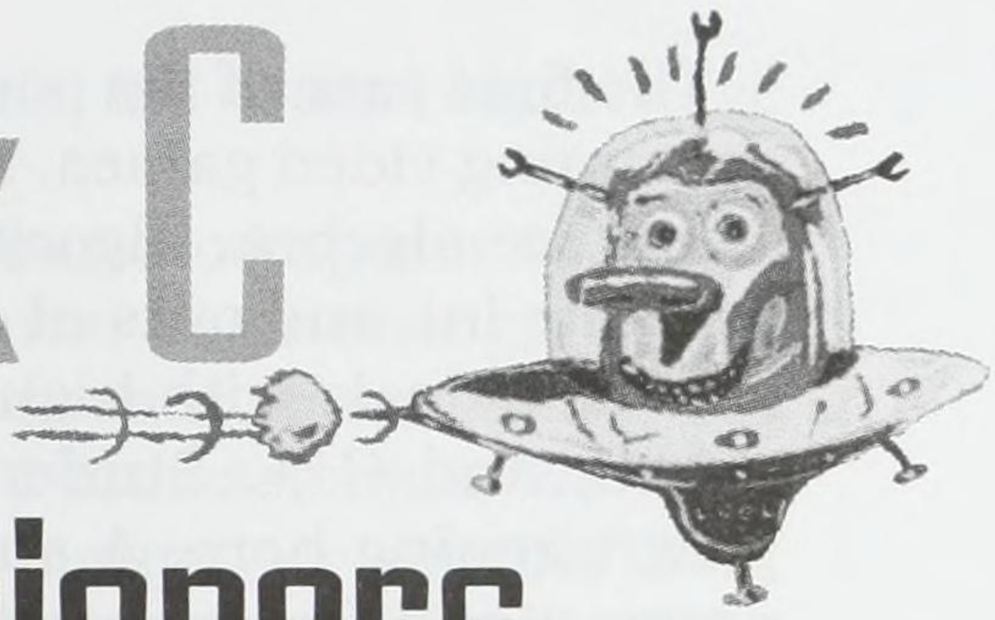
Time Warner Interactive
Mike Guttentag, Director of New Business
2210 W. Olive Avenue
Burbank, CA 91506-2626
Video/Multimedia/Computer

U.S. Gold, Inc.
Annie Uzdavinis, Marketing Manager
303 Sacramento Street, 4th Floor
San Francisco, CA 94111
Video/Multimedia/Computer

Viacom New Media
Tom Ascheim, Vice President of New Business Development
648 S. Wheeling Road
Wheeling, IL 60090
Video/Multimedia

Virgin Interactive Entertainment
Christina Camerota, Coordinator for Software Publishing
18061 Fitch Ave
Irvine, CA 92714
Video/Multimedia

Appendix C



School for Designers

Apprentice designers who want more than self-guided study have to travel. Until recently, there were only a handful of such institutions, all of them in Japan. And, of course, they all require fluent Japanese. As more North American companies look for a dependable talent pool, opportunities for higher education in game design have increased.

The first video game design and programming school in the Western hemisphere is the DigiPen Applied Computer Graphics School, in Vancouver, BC, Canada. Created with help from Nintendo of America, the Applied Computer Graphics School uses the Super NES development system in a two-year course that stresses the technical and engineering aspects of interactive game design. The school plans to move on to courses centered on newer, more powerful systems, when the SNES passes into history.

According to the president of DigiPen, Claude P. Comair, the program is a condensed course in computer programming in general, and game programming in particular. It's designed to serve as a complete program or a stepping stone toward longer university programs.

The student handbook spells out the school's aims and methods. "As the end product (game programming) is well defined, the program's course contents are well focused and deal with the subjects promptly. The aim of the program is to produce experts in this field, thus there is no time to waste on 'fill-in' subjects."

The first year of the course, the Foundation Year, explores programming video games. Semester I focuses on the essential subjects like algebra, algorithms, probability, and statistics, along with the ins and outs of computer input and output devices. Semester II deals with high-level programming languages like Pascal, C, and C++. Students will also be introduced to low-level programming here. A summer workshop in practical high-level programming wraps up the first year.

To pass the first year students must complete all required assignments and classwork with an average of 50 percent (top fifty percent of the class) or more and maintain a minimum 70 percent attendance record.

The second year, the Production Year, allows students to go hands-on with creating their own video games for the Super Nintendo Entertainment System. In an unprecedented bout of generosity, Nintendo provided hardware for educational purposes to the school. Working with Super NES development systems, students will design and play their own video games. Like the first year, the Production Year is divided into two semesters and a summer workshop.

The first semester of the second year deals with low-level machine and assembly language programming, and the last semester, semester IV, is where students finally begin implementing a game on the Super NES. There is a summer workshop this year as well: It focuses on dealing with the overload from semester IV.

A little over eight hundred inquiries were made for the first class in September of 1994. Thirty students were accepted. Many applicants could not meet the challenging requirements for the school. To even be considered for acceptance, students must have maintained a B average or better in the last three years of high school math courses.

If you want to sign up for classes, better hurry. They have students registered up to the year 2000: thirteen and fourteen year olds are registered now to start classes when they graduate from high school. They will have to pass an entrance examination that will test their proficiency in English, their mathematics level, and any potential ideas of games or storyboards.

The cost is reasonable: $8,500 (Canadian) per year, which is roughly $6,000 a year in US dollars. It's projected that if Nintendo

had not donated the development systems, the course could have cost as much as $17,000 a year.

DigiPen originally specialized in advanced computer technology and services in areas like architecture, engineering, motion pictures, and advertising. "Games are becoming more and more like movies, and companies need trained professionals. We want to provide a good, official, mainstream manpower source," said Mr. Comair. He also said software companies began calling the first year to ask when the first class was due to graduate.

For admissions information, contact the DigiPen Applied Computer Graphics School, Attention Jason Chu (or) Megan Shoesmith, 5th Floor, 530 Hornsby St., Vancouver BC, V6C-2E7.

NBCC Miramichi isn't brand-new, but it's just slamming into high gear. This community college in New Brunswick, Canada, has taught courses on cutting-edge technologies for some time, but it was only in January 1995 that it began a course in electronic game design.

This school has taken a radical approach. The course immerses students in the same environment that they would encounter in real life business. "There's nothing like us in the world, period," asserts Don Whittey, Department Head. "We've been given a lot of freedom to try new things."

The Electronic Game Design Course is an eighty-week program that gives students the technical and management skills to produce an internationally marketable video game. The curriculum emphasizes design techniques, developing technologies, and business skills. Students also receive instruction in graphics, animation, product development, programming, and gameplay. The curriculum also emphasizes the ability to work as part of a team.

The school's stated goal is to make entrepreneurs. In order to graduate, each student must design, program, and document a marketable electronic game. Students must build design teams in concert with fellow students. Other courses offered include artificial intelligence technology, virtual reality technology, multimedia technology, and educational technology.

Student design teams can be made from any combination of the various disciplines; the other courses also require students to have created a viable product (in their own fields) in order to graduate.

To be accepted at NBCC Miramichi, candidates must have at least a high school diploma and have shown aptitude in basic college English and mathematics. NBCC turns down 40 to 50 percent of the applicants.

The first class ranges in age from nineteen to fifty-seven; twenty-five of them are directly out of high school. Students attend a half day of classes and work on their project the rest of the time.

There are no exams, but don't think that means it's easy. At the end of the first three months, students must have a design company, a business plan, and the basic design for a marketable product that has been approved by a software publisher or a major market survey.

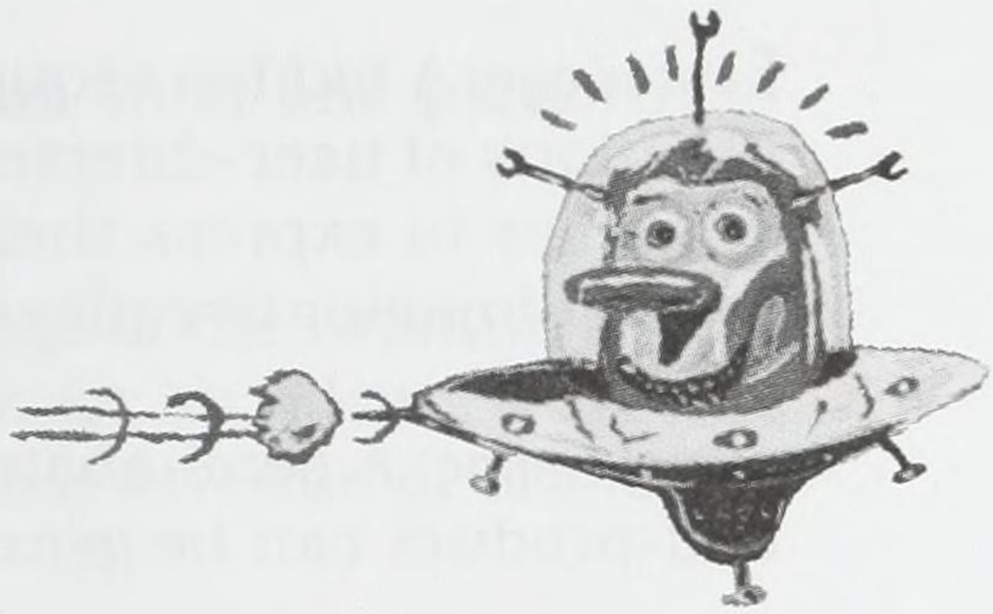
Each group has a staff advisor, but the students come up with their own ideas and do the legwork to assemble their own design teams. Course content includes a very eclectic and wide range of skills, including Historical Research (Games); Game Design I, II & III; CD/Laser Disk Production; Audio Compression Techniques; Introduction to Game Graphics; Artificial Intelligence and Embedded AI Techniques; Object Based Programming; Product Development; Suspension of Disbelief; Interactive Characters and Dialogue; Marketing and Venture Capital; among others too numerous to name here.

"The tuition," says Don Whittey, "is embarrassingly cheap." The course itself is $800 (Canadian) a year, while books and supplies cost $1,000 (Canadian).

Students learn primarily on Pentium platforms, with SoftImage programs and Silicon Graphics Indigo units used on some graphics applications. They will do some development on Macintosh, but intend to aim more for broad multimedia development.

For admissions, contact NBCC Miramichi, Attention Laurie Jean Wallace, 80 University Avenue, P.O. Box 1053, Chatham, New Brunswick, Canada, E1N 3W4.

Glossary



Game design has also borrowed terms from many disciplines, especially from films, and sometimes redefined them. Some call noninteractive sequences in games "out-takes," but filmmakers reserve that word to mean versions of a scene that aren't used in the final cut of a film.

To help you understand some of these references, we present this short dictionary of game design terms.

activity toy: Also called a "software toy," a nongame leisure product centered on undirected play. The user derives entertainment from the manipulation of product elements, since there are no victory conditions or winners.

alpha stage: The first prepublication stage of a game. Enough elements of the game have been combined to allow for testing and fine-tuning. Some major elements, such as music, are frequently missing, and some games may not have all their levels installed.

beta stage: The last version of a game before release. The game is ready for in-house evaluation and quality assurance by people outside the development group. Publishers frequently authorize small changes to a beta, but the product is generally at least 90 percent complete.

coin-op: A commercial amusement device, usually found in arcades and other public places, which activates when the customer deposits a coin, token, or other form of payment in the machine's receptacle.

Easter egg: A hidden sequence or image, usually triggered by combinations of user-directed actions. Designers embed Easter eggs in games to express their individuality and, more frequently, to reward imaginative players.

game engine: A semi-abstract program with which more than one end-product can be generated.

implementation: The translation of a game design from the specifications document to interactive media.

interface: The keyboard commands and other control devices through which a user directs the action of a game.

license: A name, logo, likeness, or intellectual property that is bought by a publisher to serve as the basis of a game. Some games, including many sports titles, have two or more licenses to lend authenticity.

milestone payment: Scheduled remittances to designers and others involved in the creation of the product. Publishers break up the advance into milestone payments because it provides incentive for completing each stage of the product.

platform: A hardware system. It is so-named because it supports the development and marketing of software.

playfield: The game screen on which the major game interactions take place.

play-life: How many hours the typical buyer will enjoy playing a game.

play-mechanics: The structure and procedures of play; what could be called the "rules" of the game.

port (n.): A product developed for one system that is extended to another with no substantial changes.

port (v.): The act of transfering a program from one platform to another.

royalties: A payment based on a percentage of the revenue generated by a game. Designers usually work for an advance against royalties based on a percentage of the net revenue or a specific payment per unit sold.

shooters: A game genre in which the player fires a range weapon at a target.

simulation: An interactive entertainment product that emulates real-world activities, such as flying a plane; these can be competitive or noncompetitive.

SKU: An abbreviation for stock-keeping unit, which refers to each platform or version of a title in distribution. One SKU means there is only one version of a game (it operates on a single platform); two SKUs refers to two versions, and so on. In popular parlance, "SKU work" is translating an already designed game into a form usable on another platform.

specifications document: The complete written description of the design of an electronic gaming product. It explains every facet of a proposed game, and it is used as the blueprint for producing the final game.

vanity board: From the arcades, a game screen that presents accumulated high score information.

zero-sum game: An interactive contest in which there must be a winner and a loser.

Index

A

Access Software, Inc., 249
Acclaim Entertainment, Inc., 249
Accolade, Inc., 249
Activision
  address and contact, 250
  in early game design, 18-20
Activity toy, definition, 267
Adams, Scott, in early game design, 22
Adults, effect on game revival, 55-57
Adventure, as early game, 21-22
Agents, role in game proposals, 111
Alliance Interactive Software, Inc., 250
Alpha stage, definition, 267
American Laser Games, 250
American Sammy, 250
American Technos, Inc., 250
AMTEX Software Corporation, 250
Animations, final design, 145
APBA Baseball, design standards and, 77-78
Apple II, early games and, 25
Aris Entertainment, 251
Art, commercialism and, in game design, 72-73
ASG Technologies, Inc., 251
Atari
  address and contact, 251
  game history and, 1, 3, 5, 7-8, 13, 15-20, 25-26, 29, 31, 34, 39-40, 43, 186-187, 190
Atlus Software, Inc., 251
Audio, in game design, 88
Avalon Hill Game Company, 251

B

Baer, Ralph H., electronic game history and, 5, 9-11
Baker & Taylor, Inc., 251
Bally Professional Arcade, 13-16
BAO Simulation Products, 252
Barnett, Ted, 215-218
Berlyn, Mike, 153-155
Beta stage, definition, 267
Blank, Marc, in early game design, 24
British games, design standards and, 75-76
British software, role in game revival, 50-52
Broderbund Software, Inc., 252
Bubsy, 153
Bullet-Proof Software, 252
Bullfrog Productions, 172-174
Bushnell, Nolan, role in game development, 5, 7-8, 10

C

Cambrix, Publishing, 252
Capcom U.S.A., 252
Capstone Software, 252
Careers, in game design, 219-248
Carlson, Julie, 155-160
Celtic Tales, 156, 158, 168, 170
Channel F, in early game history, 12-13, 15-16
Chardot, Hubert, 202-205
Charpenet, Patrick, 202-205
Checkmate, as early game, 13-14
Clayfighter, 192
Claymates, 192
Coin-op, definition, 267
Coleco, early game design and, 34-35, 44
ColecoVision, role in game development, 34-35
Coleridge, Samuel Taylor, game design and, 75-76

College program, for game designer, 222-225
Commanche Overkill, as simulator, 67-68
Commercialism, art and, in game design, 72-73
Commodore Business Machines, 47-50
Commodore 64, effect on game development, 48-52
Commodore VIC-20, 25-26
Compton's New Media, Inc., 253
Computer equipment, for game
Computer graphics, design career and, 224
Computer science courses, for designers, 223
Concept, as key to game design, 82-84, 99-107
Construction Kit
final design, 148
preliminary design and, 129-130
in sample design proposal, 246-247
Cover letters, 232
Crane, David, 18-20
Creative Multimedia Corporation, 253
Crum, Bill, 209-211
Crystal Dynamics, 253
Cyan games, 199

D

Davidson & Associates, Inc., 253
Deadly Games/Chaos Tech, 253
Desert Strike, as simulation, 67-68
Design, for games. *See* Electronic game design.
Design proposal. *See* Proposals.
Design specifications
document, definition, 269
final, 133-150
preliminary, 119-131
Development/design teams, 239-240
Development houses, 231
DigiPen Applied Computer Graphics School, 263-265
Disney Software, 253
Document authorship, design career and, 234
Dongleware, 254
Draftsmanship, design career and, 224

E

Easter egg, definition, 268
Electro Brain Corp., 254
Electronic Arts, 27-28, 92, 178
address, 254
Electronic game design
agents and, 111
careers in, 219-248
computer equipment and, 110
conceptualization as key to, 82-84, 99-107
future hardware for, 236-238
future markets, 235-236
history, 1-58
laundry lists and, 111-115
project organization, 95-96
proposals for, 109-118
schools, 263-266
standards, 71-79
teams, future of, 239-240
themes as key to, 82-84
theories behind, 59-98
United States standards, 75-79
Electronic game designers
career, getting started, 229-234
career, preparation for, 221-227
college program for, 222-225
definition, 1-2
document authorship and, 234
first design job, 230-232
history, 92-98
inspiration and, 102-106
job description, 91-98
journalism and, 233-234
list of, 151-218
as part of team, 92, 95-98
prospective employers, 230-234
quality assurance jobs, 234
roots of inspiration, 103-106

Electronic games
anatomy of, 81-90
concept development, 82-84
design for, history, 1-58
elements, 63-65
golden age, 15-36
the great crash, 37-45
history of, 1-58
in real time, 87-88
renaissance, 47-58
simulations and, comparison, 65-68
themes as essential to, 82-84
turn-based, 87-88
Electronic Games, 224-225
Electronic Gaming Fandom, design career and, 225-226
Electronic Gaming Monthly, 183
English skills, design career and, 223
Entrepreneurial start-ups, 231-232

F

The Fall and Rise of Video Games, 224
FCI, 254
Final design
animations, 145
format, 134
interface, 145
main display and, 141-143
music, 146
object list, 143-145
outline for, 135
sound, 146
specifications, 133-150
theme, 135
timetable operation, 148
First Star Software, Inc., 254
Flight Simulator 5, 65
Freeway, as early video game, 19
Friedman, Barry, 30
Fulop, Rob, 215-218

G

Game engine, definition, 268
Game Over, 224
Games
definition, 62
electronic. *See* Electronic games.
traditional and electronic, comparison, 62-63
Gametek, Inc., 254
Genghis Khan, 156-157
Goro, Phil, 160-163
Graphics
final design and, 145
in game design, 88-89
preliminary design and, 126-127
in sample design proposal, 244
Grossman, Dave, 205-209
Gunfight, as early game, 13-14
Gush, Rick, 163-168

H

Hackers, electronic games and, 4-5
Hardware, for design in future, 236-238
Hawkins, Trip, in early games, 27, 92
History, electronic game design, 1-58
Hi Tech Entertainment, 255
Hogue, Bill, history of game design and, 29-30
Hudson Soft U.S.A., Inc., 255
Humongous Entertainment, 255

I

I-Motion, Inc., 202, 255
Implementation, definition, 268
Impressions Software, 255
Infocom, in early game design, 24
Infogrames, 202-203
Inspiration, as key to game design, 103-106
Intellivision, role in game development, 32-34
Interface
definition, 268
final design and, 146
preliminary design and, 128-129
user, in game design, 85-87

Interplay Productions, 255
IVI Publishing, 256

J

Jasmine Multimedia Publishing, 256
Johnson, David, 168-172
Journalism, design career and, 233-234
JVC Musical Industries, Inc., 256

K

Kaboom, as early game, 20
Kemco America, Inc., 256
Kirby, Keith, 215-218
Knowledge Adventure, Inc., 256
Koei Corporation, 256
Kotok, Alan, role in game history, 5-6

L

Landeros, Rob, 211-213
Laundry lists, for publisher review, 111-115
Learning Company, 257
Levy, James, role in video game design, 18-19
Library skills, and design career, 224
Licenses
  definition, 268
  in game design, 83-84
Logic courses, for designers, 223
LucasArts, 206-207, 257
Luminaria, 257

M

Magnavox, game history and, 10-11
Main display, final design and, 141-143
Mainframes, and early game history, 21-22
Mandingo Entertainment, Inc., 257
Marketing knowledge, design career and, 224
Marketing perspective, sample design proposal, 247
Math courses, for designers, 223
Mattel, in development of games, 32-34
Maxis, 257
Merit Software, Inc., 257
MicroLeague Baseball, design standards and, 77-78
Microleague Interactive Software, 258
MicroProse Software, Inc., 258
Microsoft, 258
Milestone payment, 268
Miller, Alan, 18-20
Miller, Rand, 198-202
Miller, Robyn, 198-202
Mindscape, 258
Miner 2049er, as early game, 29-30
MIT, early games and, 5-7
Modems, future design and, 238-239
Molyneux, Peter, 172-177
Monopoly, game design standards and, 76-77
Morph's Outpost, 225
Multimedia, for future design, 238-239
Music, final design, 146
Myst, 198-201

N

NBCC Miramichi College, 265-266
NES. *See* Nintendo.
Networking, for design career, 226
New World Computing, Inc., 258
1960s, electronic games during, 4-5
Nintendo, role in revival of games, 53-57
Nintendo of America, 258
North American Philips, in early game history, 16-19, 31

O

Object list, final design, 143-145
Ocean of America, 259

Odyssey, as early game, 1, 11-12, 15-19, 43
Opening sequence
  for final design, 136-138
  for preliminary design, 124-126
  sample design proposal, 243
Orr, Scott, 178-183
Outlines
  for final design,
  for preliminary design, 123
Out-takes, definition, 61

P

Pac-Man, and decline of early games, 39-40
Peters, Jeff, 183-186
Platform, definition, 268
Play-action, in sample design proposal, 244-245
Playfield, definition, 268
Play-life, definition, 268
Playmates Toys, Inc., 259
Play-mechanics
  definition, 87, 268
  final design, 147
  preliminary design and, 129
  sample design proposal, 245-246
Pong, as early game, 9-10
Port, definition, 61, 268-269
Preliminary design specifications, 119-131
  outline for, 123
  product overview and, 123-124
Preliminary design specifications document, 121-131
Preplay sequence, for game design, 84-85
Product overview
  final design and, 135
  preliminary design and, 123-124
  sample design proposal, 242-243
Project organization, in game development, 95-96
Proposals
  for games, 109-118
  sample, 241-248
  submission to publishers, 232-233
Psygnosis Limited, 259
Puzzles, comparison with games, 68-69

Q

Quality assurance jobs, designers and, 234
Quantum Quality Productions, 259

R

RCA, early games and, 12
Royalties, definition, 269
Russell, Steve, role in game history, 5-7

S

Sanctuary Woods, 259
Sanders and Associates, impact on game design, 8-11
Schafer, Tim, 205-209
Schools, for designers, 263-266
Sega
  address, 259
  effect on game revival, 54-56
Shooters, definition, 269
SimCity, 197-198
SimWorld, 198
Simulations, comparison with games, 65-68
Simulators, definition, 65, 67-68
Sir-tech Software, 259
Skruch, John, 186-191
SKU, definition, 269
SKU house, definition, 61
Smurf Rescue, as early game, 35-36
Software, for future design, 238-239
Software piracy, 226
Software publishers, 230
  listing, 249-262

Software Sorcery, 260
Software Toolworks, 260
Sound
final design and, 146
in game design, 88
preliminary design and, 127-128
in sample design proposal, 244
Space War, as first computer game, 4-5, 7
Specifications. *See* Design specifications.
Specifications document, definition, 269
Square Soft, Inc., 260
Strategic Simulations, Inc., 260
Street Fighter II, 191-192
Suncom Technologies, 260
Sunsoft, 260

T

Taito America Corporation, 260
Takara U.S.A. Corporation, 261
Take 2 Interactive Software, 261
Teamwork
design/development, future of, 239-240
game designers and, 92, 95-98
Tecmo, inc., 261
Telecommunications, future design and, 237
Tempest, 190
Tetragon, Inc., 261
Tetris, 64
Themes, as key to game design, 82-84
Theories, behind game design, 59-98
Thomas, Jeff, 191-193
Three-dimensional images, 237-238
Timetable operation, final design and, 148
Time Warner Interactive, 261
Tribolyte, 211-213
Tunnell, Jeff, 193-197

U

User interface, in game design, 85-87
U.S. Gold, Inc., 261
Utilities, in development of games, 89
Utility screens, final design and, 138-141

V

Vanity board, definition, 269
Viacom New Media, 262
Virgin Interactive Entertainment, 262
Virtual reality, design and, 238

W

Walls, Jim, 213-215
Westwood Studios, 160, 163, 209-210
Williams, Ken, early game design and, 23
Wired, 225
Wizardry: Proving Grounds of the Mad Overlord, 24-25
Wolfenstein 3D, 65
World War II, impact on electronic game design, 3
Wright, Will, 197-198

Z

Zero-sum game, 269

U.S. $24.95/Can. $36.95/U.K. £22.99 Net

Electronic Entertainment/Business

Become a Game Designer!

Learn the Secrets of the Trade from Professionals

Learn how to take a raw concept for a terrific game and turn it into a design specifications document even seasoned professionals will respect. This book covers every step! Starting with a history of electronic game design, *Inside Electronic Game Design* guides you through everything you need to know to crack the game design industry.

Find out what goes on behind the scenes in the exciting world of interactive electronic entertainment! Here at last is a complete overview of how the game business works, what the game designer's role is, and how you can become a serious player. From dream to disk, this book covers everything you need to know to start a career in game design! Learn how to:

- Explore careers in the game design industry
- Create a red-hot proposal
- Develop the preliminary design
- Network like an insider
- And much more!

EXCLUSIVE Interviews with over 24 leading game designers EXCLUSIVE

Including a sample game proposal and lists of contacts and schools for designers, this book and your imagination are all you need to plunge into the fast-paced world of game design.

ARNIE KATZ, president of Katz & Worley Inc., has participated in the design of nearly two dozen games, including Batman Returns!, MicroLeague Baseball, and Electronic Blood Bowl. He is a product consultant to many computer and video game publishers.

LAURIE YATES, a writer with Fog Studios, is co-author of *Buried in Time: The Journeyman Project 2—The Official Strategy Guide* (Prima) and co-editor of the "Cyberwatch" column for *Fusion* magazine.

Prima Publishing

ISBN 1-55958-669-9

9 781559 586696 52495 >